Algebra, Logic and Complexity in Celebration of Eric Allender and Mike Saks

Neil Immerman

College of Computer and Information Sciences
University of Massachusetts, Amherst
Amherst, MA, USA

`people.cs.umass.edu/~immerman`

31 years ago, STOC and Structures in Berkeley.

Reductions

- [C71] SAT is NP complete via ptime Turing reductions.

Reductions

- [C71] SAT is NP complete via ptime Turing reductions.
- [K72] Many important problems are NP complete, via $\leq_p$.

Reductions

- [C71] SAT is NP complete via ptime Turing reductions.
- [K72] Many important problems are NP complete, via $\leq_{\mathrm{p}}$.
- [J73] ... stay complete via **logspace** reductions, $\leq_{\log}$.

Reductions

- [C71] SAT is NP complete via ptime Turing reductions.
- [K72] Many important problems are NP complete, via $\leq_{\mathrm{p}}$.
- [J73] ... stay complete via **logspace** reductions, $\leq_{\mathrm{log}}$.
- [HIM78] ... stay complete via **one-way logspace**, reductions, $\leq_{\text{1-log}}$.

Reductions

- [C71] SAT is NP complete via ptime Turing reductions.
- [K72] Many important problems are NP complete, via $\leq_{\mathrm{p}}$.
- [J73] ... stay complete via **logspace** reductions, $\leq_{\mathrm{log}}$.
- [HIM78] ... stay complete via **one-way logspace**, reductions, $\leq_{\text{1-log}}$.
- [I80] ... stay complete **first-order** reductions.

Reductions

- [C71] SAT is NP complete via ptime Turing reductions.
- [K72] Many important problems are NP complete, via $\leq_{\mathrm{p}}$.
- [J73] ... stay complete via **logspace** reductions, $\leq_{\log}$.
- [HIM78] ... stay complete via **one-way logspace**, reductions, $\leq_{\text{1-log}}$.
- [I80] ... stay complete **first-order** reductions.
- [V82] ... stay complete via **projections**. (Non-uniform reductions where each bit of the output depends on at most one bit of the input).

Reductions

- [C71] SAT is NP complete via ptime Turing reductions.
- [K72] Many important problems are NP complete, via $\leq_{\mathrm{p}}$.
- [J73] … stay complete via **logspace** reductions, $\leq_{\mathrm{log}}$.
- [HIM78] … stay complete via **one-way logspace**, reductions, $\leq_{\mathrm{1\text{-}log}}$.
- [I80] … stay complete **first-order** reductions.
- [V82] … stay complete via **projections**. (Non-uniform reductions where each bit of the output depends on at most one bit of the input).
- [I87] … stay complete via **first-order projections**, $\leq_{\mathrm{fop}}$.

Reductions

- [C71] SAT is NP complete via ptime Turing reductions.
- [K72] Many important problems are NP complete, via $\leq_{\mathrm{p}}$.
- [J73] ... stay complete via **logspace** reductions, $\leq_{\mathrm{log}}$.
- [HIM78] ... stay complete via **one-way logspace**, reductions, $\leq_{\text{1-log}}$.
- [I80] ... stay complete **first-order** reductions.
- [V82] ... stay complete via **projections**. (Non-uniform reductions where each bit of the output depends on at most one bit of the input).
- [I87] ... stay complete via **first-order projections**, $\leq_{\mathrm{fop}}$.
- [L75] Artificial, non-complete problems can be constructed.

Reductions

- [C71] SAT is NP complete via ptime Turing reductions.
- [K72] Many important problems are NP complete, via $\leq_{\mathrm{p}}$.
- [J73] ...stay complete via **logspace** reductions, $\leq_{\mathrm{log}}$.
- [HIM78] ...stay complete via **one-way logspace**, reductions, $\leq_{\text{1-log}}$.
- [I80] ...stay complete **first-order** reductions.
- [V82] ...stay complete via **projections**. (Non-uniform reductions where each bit of the output depends on at most one bit of the input).
- [I87] ...stay complete via **first-order projections**, $\leq_{\mathrm{fop}}$.
- [L75] Artificial, non-complete problems can be constructed.
- **Dichotomy:** "Natural" problems are complete for important compexity classes [FV99, S78, ABISV09].

Isomorphism Conjecture

- [BH77] **Isomorphism Conjecture:** "All NP complete sets via ptime many-one reductions, $\leq_{\mathrm{p}}$, are polynomial-time isomorphic."

Isomorphism Conjecture

- [BH77] **Isomorphism Conjecture:** "All NP complete sets via ptime many-one reductions, $\leq_p$, are polynomial-time isomorphic."
- [ABI93] **fop Isomorphism Thm.** All NP complete sets via $\leq_{fop}$ are first-order isomorphic. Also true for L, NL, P, PSPACE, etc.

fop Isomorphism Theorem proof sketch

[BH77] **Observation:** All the NP complete sets in [GJ] are p-isomorphic.

fop Isomorphism Theorem proof sketch

[BH77] **Observation:** All the NP complete sets in [GJ] are p-isomorphic.

Schröder-Bernstein Thm. Let A and B be any two sets and suppose that $f : A \overset{1:1}{\to} B$ and $g : B \overset{1:1}{\to} A$. Then there exists $h : A \underset{\text{onto}}{\overset{1:1}{\to}} B$.

fop Isomorphism Theorem proof sketch

[BH77] **Observation:** All the NP complete sets in [GJ] are p-isomorphic.

Schröder-Bernstein Thm. Let A and B be any two sets and suppose that $f : A \overset{1:1}{\to} B$ and $g : B \overset{1:1}{\to} A$. Then there exists $h : A \underset{\text{onto}}{\overset{1:1}{\to}} B$. $\quad (|A| \leq |B| \;\wedge\; |B| \leq |A| \;\to\; |A| = |B|)$

fop Isomorphism Theorem proof sketch

[BH77] **Observation:** All the NP complete sets in [GJ] are p-isomorphic.

Schröder-Bernstein Thm. Let A and B be any two sets and suppose that $f : A \overset{1:1}{\to} B$ and $g : B \overset{1:1}{\to} A$. Then there exists $h : A \underset{\text{onto}}{\overset{1:1}{\to}} B$. $\quad (|A| \leq |B| \;\wedge\; |B| \leq |A| \;\to\; |A| = |B|)$

Proof: For $a, c \in A \cup B$, say that a is an **ancestor** of c if we can go from a to c by applying a finite, non-zero, number of applications of f and g.

fop Isomorphism Theorem proof sketch

[BH77] **Observation:** All the NP complete sets in [GJ] are p-isomorphic.

Schröder-Bernstein Thm. Let A and B be any two sets and suppose that $f : A \overset{1:1}{\to} B$ and $g : B \overset{1:1}{\to} A$. Then there exists $h : A \underset{\text{onto}}{\overset{1:1}{\to}} B$. $\quad (|A| \le |B| \ \wedge \ |B| \le |A| \ \to \ |A| = |B|)$

Proof: For $a, c \in A \cup B$, say that a is an **ancestor** of c if we can go from a to c by applying a finite, non-zero, number of applications of f and g.

$$h(a) \stackrel{\text{def}}{=} \begin{cases} g^{-1}(a) & \text{if } a \text{ has an odd number of ancestors} \\ f(a) & \text{if } a \text{ has an even or infinite number of ancestors} \end{cases}$$

fop Isomorphism Theorem proof sketch

[BH77] **Observation:** All the NP complete sets in [GJ] are p-isomorphic.

Schröder-Bernstein Thm. Let A and B be any two sets and suppose that $f : A \overset{1:1}{\to} B$ and $g : B \overset{1:1}{\to} A$. Then there exists $h : A \underset{\text{onto}}{\overset{1:1}{\to}} B$. $\qquad (|A| \leq |B| \wedge |B| \leq |A| \to |A| = |B|)$

Proof: For $a, c \in A \cup B$, say that a is an **ancestor** of c if we can go from a to c by applying a finite, non-zero, number of applications of f and g.

$$h(a) \stackrel{\text{def}}{=} \begin{cases} g^{-1}(a) & \text{if } a \text{ has an odd number of ancestors} \\ f(a) & \text{if } a \text{ has an even or infinite number of ancestors} \end{cases}$$

Thus, $h : A \underset{\text{onto}}{\overset{1:1}{\to}} B$ □

[BH77] **Observation:** All the NP complete sets in [GJ] are p-isomorphic.

[BH77] **Observation:** All the NP complete sets in [GJ] are p-isomorphic.

Lemma: Let $f : A \leq_p B$ and $g : B \leq_p A$ where f and g are 1:1 length-increasing functions. Assume also that f and g have left inverses in FP. Then A is p-isomorphic to B.

[BH77] **Observation:** All the NP complete sets in [GJ] are p-isomorphic.

Lemma: Let $f : A \leq_{\mathrm{p}} B$ and $g : B \leq_{\mathrm{p}} A$ where f and g are 1:1 length-increasing functions. Assume also that f and g have left inverses in FP. Then A is p-isomorphic to B.

Proof: Since f, g are length-increasing, the ancestor chains are linear in length. Thus, the isomorphism, h, can be defined as in the SB Thm, but now it can be computed in ptime. □

Def. $A \subseteq \Sigma^*$ has **p-time padding functions** if $\exists e, d \in \mathrm{FP}$ s.t.

1. $\forall w, x \in \Sigma^* \quad w \in A \leftrightarrow e(w, x) \in A$
2. $\forall w, x \in \Sigma^* \quad d(e(w, x)) = x$
3. $\forall w, x \in \Sigma^* \quad |e(w, x)| \geq |w| + |x|$.

Def. $A \subseteq \Sigma^*$ has **p-time padding functions** if $\exists e, d \in \mathrm{FP}$ s.t.

1. $\forall w, x \in \Sigma^* \quad w \in A \leftrightarrow e(w, x) \in A$
2. $\forall w, x \in \Sigma^* \quad d(e(w, x)) = x$
3. $\forall w, x \in \Sigma^* \quad |e(w, x)| \geq |w| + |x|$.

Example: for SAT: $\quad e(w, x) \stackrel{\text{def}}{=} (w) \wedge \underbrace{C_1 \wedge \cdots \wedge C_{|x|}}$, where

$C_i = (y \vee \overline{y})$ if $x_i = 1$, else $(\overline{y} \vee y)$.

Def. $A \subseteq \Sigma^*$ has **p-time padding functions** if $\exists e, d \in \mathrm{FP}$ s.t.

1. $\forall w, x \in \Sigma^* \quad w \in A \leftrightarrow e(w,x) \in A$
2. $\forall w, x \in \Sigma^* \quad d(e(w,x)) = x$
3. $\forall w, x \in \Sigma^* \quad |e(w,x)| \geq |w| + |x|$.

Example: for SAT: $e(w,x) \stackrel{\text{def}}{=} (w) \wedge \underbrace{C_1 \wedge \cdots \wedge C_{|x|}}$, where $C_i = (y \vee \overline{y})$ if $x_i = 1$, else $(\overline{y} \vee y)$.

Lemma: If $A, B \in \mathrm{NPC}$ and have p-time padding functions, then they are inter-reducible via p-time invertible 1:1 length-increasing reductions.

Def. $A \subseteq \Sigma^*$ has **p-time padding functions** if $\exists e, d \in \mathrm{FP}$ s.t.

1. $\forall w, x \in \Sigma^* \quad w \in A \ \leftrightarrow \ e(w,x) \in A$
2. $\forall w, x \in \Sigma^* \quad d(e(w,x)) = x$
3. $\forall w, x \in \Sigma^* \quad |e(w,x)| \geq |w| + |x|$.

Example: for SAT: $e(w,x) \stackrel{\text{def}}{=} (w) \wedge \underbrace{C_1 \wedge \cdots \wedge C_{|x|}}$, where

$C_i = (y \vee \overline{y})$ if $x_i = 1$, else $(\overline{y} \vee y)$.

Lemma: If $A, B \in \mathrm{NPC}$ and have p-time padding functions, then they are inter-reducible via p-time invertible 1:1 length-increasing reductions.

Lemma: All the NP complete sets in [GJ] have p-time padding functions.

Def. $A \subseteq \Sigma^*$ has **p-time padding functions** if $\exists e, d \in \mathrm{FP}$ s.t.

1. $\forall w, x \in \Sigma^* \quad w \in A \ \leftrightarrow \ e(w,x) \in A$
2. $\forall w, x \in \Sigma^* \quad d(e(w,x)) = x$
3. $\forall w, x \in \Sigma^* \quad |e(w,x)| \geq |w| + |x|$.

Example: for SAT: $\quad e(w,x) \stackrel{\text{def}}{=} (w) \wedge \underbrace{C_1 \wedge \cdots \wedge C_{|x|}}$, where $C_i = (y \vee \overline{y})$ if $x_i = 1$, else $(\overline{y} \vee y)$.

Lemma: If $A, B \in \mathrm{NPC}$ and have p-time padding functions, then they are inter-reducible via p-time invertible 1:1 length-increasing reductions.

Lemma: All the NP complete sets in [GJ] have p-time padding functions.

Thus, all the NP complete sets in [GJ] are p-isomorphic. $\square$

fop Isomorphism Theorem proof sketch

fop Isomorphism Thm. All NP complete sets via $\leq_{\text{fop}}$ are first-order isomorphic. Also true for NC^1, sAC^1, L, NL, P, PSPACE, etc.

fop Isomorphism Theorem proof sketch

fop Isomorphism Thm. All NP complete sets via $\leq_{\text{fop}}$ are first-order isomorphic. Also true for NC^1, sAC^1, L, NL, P, PSPACE, etc.

Key Lemma: Let f be a first-order projection (fop) that is 1:1 and of arity at least 2, i.e., it at least squares the size. Then the following two predicates are first-order expressible concerning a structure, $\mathcal{A}$:

1. IE($\mathcal{A}$), meaning that $f^{-1}(\mathcal{A})$ exists.
2. #Ancestors($\mathcal{A}, r$), meaning $\mathcal{A}$ has exactly r ancestors.

fop Isomorphism Theorem proof sketch

fop Isomorphism Thm. All NP complete sets via $\leq_{\text{fop}}$ are first-order isomorphic. Also true for NC^1, sAC^1, L, NL, P, PSPACE, etc.

Key Lemma: Let f be a first-order projection (fop) that is 1:1 and of arity at least 2, i.e., it at least squares the size. Then the following two predicates are first-order expressible concerning a structure, $\mathcal{A}$:

1. IE($\mathcal{A}$), meaning that $f^{-1}(\mathcal{A})$ exists.
2. #Ancestors($\mathcal{A}, r$), meaning $\mathcal{A}$ has exactly r ancestors.

The rest of the proof is similar to proof from [BH77]. □

fop Isomorphism Thm. For nice complexity classes, all complete sets via fops are first-order isomorphic.

fop Isomorphism Thm. For nice complexity classes, all complete sets via fops are first-order isomorphic.

- **Morally**, the **BH Isomorphism Conjecture** is **true**.

fop Isomorphism Thm. For nice complexity classes, all complete sets via fops are first-order isomorphic.

- **Morally**, the **BH Isomorphism Conjecture** is **true**.
- Each **nice complexity class** has exactly **one complete problem**.

fop Isomorphism Thm. For nice complexity classes, all complete sets via fops are first-order isomorphic.

- **Morally**, the **BH Isomorphism Conjecture** is **true**.
- Each **nice complexity class** has exactly **one complete problem**.
- **Dichotomy Phenomenon: "Natural"** computational problems tend to be **complete via fops** for one of our **favorite complexity classes**.

fop Isomorphism Thm. For nice complexity classes, all complete sets via fops are first-order isomorphic.

- **Morally**, the **BH Isomorphism Conjecture** is **true**.
- Each **nice complexity class** has exactly **one complete problem**.
- **Dichotomy Phenomenon:** **"Natural"** computational problems tend to be **complete via fops** for one of our **favorite complexity classes**.
- Great for **Algorithms** and **Complexity Theory**!

fop Isomorphism Thm. For nice complexity classes, all complete sets via fops are first-order isomorphic.

- **Morally**, the **BH Isomorphism Conjecture** is **true**.
- Each **nice complexity class** has exactly **one complete problem**.
- **Dichotomy Phenomenon: "Natural"** computational problems tend to be **complete via fops** for one of our **favorite complexity classes**.
- Great for **Algorithms** and **Complexity Theory**!
- But not true in general [L75].

fop Isomorphism Thm. For nice complexity classes, all complete sets via fops are first-order isomorphic.

- **Morally**, the **BH Isomorphism Conjecture** is **true**.
- Each **nice complexity class** has exactly **one complete problem**.
- **Dichotomy Phenomenon: "Natural"** computational problems tend to be **complete via fops** for one of our **favorite complexity classes**.
- Great for **Algorithms** and **Complexity Theory**!
- But not true in general [L75].
- Why does this seem to occur?

fop Isomorphism Thm. For nice complexity classes, all complete sets via fops are first-order isomorphic.

- **Morally**, the **BH Isomorphism Conjecture** is **true**.
- Each **nice complexity class** has exactly **one complete problem**.
- **Dichotomy Phenomenon: "Natural"** computational problems tend to be **complete via fops** for one of our **favorite complexity classes**.
- Great for **Algorithms** and **Complexity Theory**!
- But not true in general [L75].
- Why does this seem to occur?
- **Logical** and **Algebraic** reasons, e.g., CSP.

fop Isomorphism Thm. For nice complexity classes, all complete sets via fops are first-order isomorphic.

fop Isomorphism Thm. For nice complexity classes, all complete sets via fops are first-order isomorphic.

Prop. [ABI93] SAT is NP complete via fops. There is a set S which is NP complete via uniform NC^0 reductions and FO isomorphic to SAT, but not NP complete via projections.

fop Isomorphism Thm. For nice complexity classes, all complete sets via fops are first-order isomorphic.

Prop. [ABI93] SAT is NP complete via fops. There is a set S which is NP complete via uniform NC^0 reductions and FO isomorphic to SAT, but not NP complete via projections.

Some were unhappy with the fop Iso Thm because of a **mismatch**: fop more restrictive than fo.

fop Isomorphism Thm. For nice complexity classes, all complete sets via fops are first-order isomorphic.

Prop. [ABI93] SAT is NP complete via fops. There is a set S which is NP complete via uniform NC^0 reductions and FO isomorphic to SAT, but not NP complete via projections.

Some were unhappy with the fop Iso Thm because of a **mismatch**: fop more restrictive than fo.

This problem is solved in [AAR96].

[AAR96]

From now on, assume every complexity classes we consider, $\mathcal{C}$, is closed under uniform NC^1 reductions.

[AAR96]

From now on, assume every complexity classes we consider, $\mathcal{C}$, is closed under uniform NC^1 reductions.

Isomorphism Thm. All sets complete for $\mathcal{C}$ under non-uniform AC^0 reductions are isomorphic under non-uniform AC^0 isomorphisms.

[AAR96]

From now on, assume every complexity classes we consider, $\mathcal{C}$, is closed under uniform NC^1 reductions.

Isomorphism Thm. All sets complete for $\mathcal{C}$ under non-uniform AC^0 reductions are isomorphic under non-uniform AC^0 isomorphisms.

Gap Thm. All sets complete for $\mathcal{C}$ under non-uniform AC^0 reductions are in fact complete under non-uniform NC^0 reductions.

[AAR96]

From now on, assume every complexity classes we consider, $\mathcal{C}$, is closed under uniform NC^1 reductions.

Isomorphism Thm. All sets complete for $\mathcal{C}$ under non-uniform AC^0 reductions are isomorphic under non-uniform AC^0 isomorphisms.

Gap Thm. All sets complete for $\mathcal{C}$ under non-uniform AC^0 reductions are in fact complete under non-uniform NC^0 reductions.

Gap Thm does not extend to uniform case. There are sets complete for $\mathcal{C}$ under FO reductions but not under fops or other uniform NC^0 reductions. (Recall FO = uniform AC^0.)

[AAR96] Isomorphism Thm proof ideas

Def. An NC^0 reduction is a **super-projection** if a subset, *S*, of its output bits is a projection s.t. each bit of its input is mapped to a bit of *S*.

[AAR96] Isomorphism Thm proof ideas

Def. An NC^0 reduction is a **super-projection** if a subset, *S*, of its output bits is a projection s.t. each bit of its input is mapped to a bit of *S*.

Lemma: Suppose *A* is hard for $\mathcal{C}$ under P-uniform NC^0 reductions. Then *A* is hard under P-uniform, 1:1 length squaring super-projections.

[AAR96] Isomorphism Thm proof ideas

Def. An NC^0 reduction is a **super-projection** if a subset, *S*, of its output bits is a projection s.t. each bit of its input is mapped to a bit of *S*.

Lemma: Suppose *A* is hard for $\mathcal{C}$ under P-uniform NC^0 reductions. Then *A* is hard under P-uniform, 1:1 length squaring super-projections.

Proof: [clever, long and complicated combinatorial surgery on some NC^0 circuits. This is where the P-uniformity comes in.

[AAR96] Isomorphism Thm proof ideas

Def. An NC^0 reduction is a **super-projection** if a subset, *S*, of its output bits is a projection s.t. each bit of its input is mapped to a bit of *S*.

Lemma: Suppose *A* is hard for $\mathcal{C}$ under P-uniform NC^0 reductions. Then *A* is hard under P-uniform, 1:1 length squaring super-projections.

Proof: [clever, long and complicated combinatorial surgery on some NC^0 circuits. This is where the P-uniformity comes in.

Thm. All sets complete for C under P-uniform NC^0 reductions are P-uniform AC^0 isomorphic.

[AAR96] Isomorphism Thm proof ideas

Def. An NC^0 reduction is a **super-projection** if a subset, *S*, of its output bits is a projection s.t. each bit of its input is mapped to a bit of *S*.

Lemma: Suppose *A* is hard for $\mathcal{C}$ under P-uniform NC^0 reductions. Then *A* is hard under P-uniform, 1:1 length squaring super-projections.

Proof: [clever, long and complicated combinatorial surgery on some NC^0 circuits. This is where the P-uniformity comes in.

Thm. All sets complete for C under P-uniform NC^0 reductions are P-uniform AC^0 isomorphic.

Follows from **Lemma** in a similar way to [ABI93].

[AAR96] Gap Thm proof ideas

Random Reduction Lemma For any AC^0 reduction computed by a family of circuits $\{C_m\}$, there exists an $a \in \mathbf{N}$ such that, for all large m of the form r^{2a}, there is a restriction τ_m which converts C_m into an NC^0 circuit, and assigns * to at least three variables in each block of length r^{2a-1}.

[AAR96] Gap Thm proof ideas

Random Reduction Lemma For any AC^0 reduction computed by a family of circuits $\{C_m\}$, there exists an $a \in \mathbf{N}$ such that, for all large m of the form r^{2a}, there is a restriction τ_m which converts C_m into an NC^0 circuit, and assigns * to at least three variables in each block of length r^{2a-1}.

Gap Thm. All sets complete for $\mathcal{C}$ under non-uniform AC^0 reductions are in fact complete under non-uniform NC^0 reductions.

[AAR96] Gap Thm proof ideas

Random Reduction Lemma For any AC^0 reduction computed by a family of circuits $\{C_m\}$, there exists an $a \in \mathbf{N}$ such that, for all large m of the form r^{2a}, there is a restriction τ_m which converts C_m into an NC^0 circuit, and assigns * to at least three variables in each block of length r^{2a-1}.

Gap Thm. All sets complete for $\mathcal{C}$ under non-uniform AC^0 reductions are in fact complete under non-uniform NC^0 reductions.

Proof: Let A be hard for $\mathcal{C}$ under AC^0 reductions. Let $B \in \mathcal{C}$. Thus, B is AC^0 reducible to A.

Goal: show B is NC^0 reducible to A.

Given: A is hard for $\mathcal{C}$ under AC^0 reductions; $B \in \mathcal{C}$,

Show: B is NC^0 reducible to A.

Given: A is hard for $\mathcal{C}$ under AC^0 reductions; $B \in \mathcal{C}$,

Show: B is NC^0 reducible to A.

Let $B'(1^k 0z) \stackrel{\text{def}}{=}$ if $(k \not| \; |z|)$: **return**(0)

$z = u_1 u_2 \ldots u_p$, blocks of k bits each

$$v_i \stackrel{\text{def}}{=} \begin{cases} 0 & \text{if } \#_1(u_i) \equiv 0 \pmod 3 \\ 1 & \text{if } \#_1(u_i) \equiv 1 \pmod 3 \\ \epsilon & \text{otherwise} \end{cases}$$

return(1) iff $v_1 \ldots v_p \in B$

Given: A is hard for $\mathcal{C}$ under AC^0 reductions; $B \in \mathcal{C}$,

Show: B is NC^0 reducible to A.

Let $B'(1^k0z) \stackrel{\text{def}}{=}$ if ($k \not| \, |z|$): **return**(0)

$z = u_1 u_2 \ldots u_p$, blocks of k bits each

$$v_i \stackrel{\text{def}}{=} \begin{cases} 0 & \text{if } \#_1(u_i) \equiv 0 \pmod 3 \\ 1 & \text{if } \#_1(u_i) \equiv 1 \pmod 3 \\ \epsilon & \text{otherwise} \end{cases}$$

return(1) iff $v_1 \ldots v_p \in B$

B' is NC^1 reducible to B, so $B' \in \mathcal{C}$.

Let $\{C_n\}$ be AC^0 circuits reducing B' to A.

Let $B'(1^k 0 z) \stackrel{\text{def}}{=}$ if ($k \not| |z|$): **return**(0)

$z = u_1 u_2 \dots u_p$, blocks of k bits each

$$v_i \stackrel{\text{def}}{=} \begin{cases} 0 & \text{if } \#_1(u_i) \equiv 0 \pmod 3 \\ 1 & \text{if } \#_1(u_i) \equiv 1 \pmod 3 \\ \epsilon & \text{otherwise} \end{cases}$$

return(1) iff $v_1 \dots v_p \in B$

Let $B'(1^k 0 z) \stackrel{\text{def}}{=}$ if $(k \not| \, |z|)$: **return**(0)

$z = u_1 u_2 \ldots u_p$, blocks of k bits each

$$v_i \stackrel{\text{def}}{=} \begin{cases} 0 & \text{if } \#_1(u_i) \equiv 0 \pmod 3 \\ 1 & \text{if } \#_1(u_i) \equiv 1 \pmod 3 \\ \epsilon & \text{otherwise} \end{cases}$$

return(1) iff $v_1 \ldots v_p \in B$

B' is NC^1 reducible to B, so $B' \in \mathcal{C}$.

Let $\{C_n\}$ be AC^0 circuits reducing B' to A.

Let $B'(1^k 0 z) \stackrel{\text{def}}{=}$ if $(k \not| \, |z|)$: **return**(0)

$z = u_1 u_2 \dots u_p$, blocks of k bits each

$$v_i \stackrel{\text{def}}{=} \begin{cases} 0 & \text{if } \#_1(u_i) \equiv 0 \pmod 3 \\ 1 & \text{if } \#_1(u_i) \equiv 1 \pmod 3 \\ \epsilon & \text{otherwise} \end{cases}$$

return(1) iff $v_1 \dots v_p \in B$

B' is NC^1 reducible to B, so $B' \in \mathcal{C}$.

Let $\{C_n\}$ be AC^0 circuits reducing B' to A.

Let $\{C_n\}$ be AC^0 circuits reducing B' to A.

Let $B'(1^k 0 z) \stackrel{\text{def}}{=}$ if ($k \not| |z|$): **return**(0)

$z = u_1 u_2 \ldots u_p$, blocks of k bits each

$$v_i \stackrel{\text{def}}{=} \begin{cases} 0 & \text{if } \#_1(u_i) \equiv 0 \pmod 3 \\ 1 & \text{if } \#_1(u_i) \equiv 1 \pmod 3 \\ \epsilon & \text{otherwise} \end{cases}$$

return(1) iff $v_1 \ldots v_p \in B$

B' is NC^1 reducible to B, so $B' \in \mathcal{C}$.

Let $\{C_n\}$ be AC^0 circuits reducing B' to A.

Let $\{C_n\}$ be AC^0 circuits reducing B' to A.

Apply the restriction which converts C_m into an NC^0 circuit, and assigns * to at least three variables in each of the n blocks, u_i.

Let $B'(1^k 0z) \stackrel{\text{def}}{=}$ if ($k \nmid |z|$): **return**(0)

$z = u_1 u_2 \ldots u_p$, blocks of k bits each

$$v_i \stackrel{\text{def}}{=} \begin{cases} 0 & \text{if } \#_1(u_i) \equiv 0 \pmod 3 \\ 1 & \text{if } \#_1(u_i) \equiv 1 \pmod 3 \\ \epsilon & \text{otherwise} \end{cases}$$

return(1) iff $v_1 \ldots v_p \in B$

B' is NC^1 reducible to B, so $B' \in \mathcal{C}$.

Let $\{C_n\}$ be AC^0 circuits reducing B' to A.

Let $\{C_n\}$ be AC^0 circuits reducing B' to A.

Apply the restriction which converts C_m into an NC^0 circuit, and assigns * to at least three variables in each of the n blocks, u_i.

Further restrict so that there is exactly one * in each block and setting of $*_i$ is the value of v_i.

Let $B'(1^k 0 z) \stackrel{\text{def}}{=}$ if $(k \not| |z|)$: **return**(0)

$z = u_1 u_2 \ldots u_p$, blocks of k bits each

$$v_i \stackrel{\text{def}}{=} \begin{cases} 0 & \text{if } \#_1(u_i) \equiv 0 \pmod 3 \\ 1 & \text{if } \#_1(u_i) \equiv 1 \pmod 3 \\ \epsilon & \text{otherwise} \end{cases}$$

return(1) iff $v_1 \ldots v_p \in B$

B' is NC^1 reducible to B, so $B' \in \mathcal{C}$.

Let $\{C_n\}$ be AC^0 circuits reducing B' to A.

Let $\{C_n\}$ be AC^0 circuits reducing B' to A.

Apply the restriction which converts C_m into an NC^0 circuit, and assigns * to at least three variables in each of the n blocks, u_i.

Further restrict so that there is exactly one * in each block and setting of $*_i$ is the value of v_i.

We have constructed an NC^0 reduction from B to A. $\square$

Consequences of Isomorphism and Gap Theorems

For nice complexity classes, all AC^0 complete sets are AC^0 isomorphic.

Consequences of Isomorphism and Gap Theorems

For nice complexity classes, all AC^0 complete sets are AC^0 isomorphic.

- **Morally**, the **BH Isomorphism Conjecture** is **true**.

Consequences of Isomorphism and Gap Theorems

For nice complexity classes, all AC^0 complete sets are AC^0 isomorphic.

- **Morally**, the **BH Isomorphism Conjecture** is **true**.
- Each **nice complexity class** has exactly **one complete problem**.

Consequences of Isomorphism and Gap Theorems

For nice complexity classes, all AC^0 complete sets are AC^0 isomorphic.

- **Morally**, the **BH Isomorphism Conjecture** is **true**.
- Each **nice complexity class** has exactly **one complete problem**.
- **Dichotomy Phenomenon: "Natural"** computational problems tend to be **complete via fops** for one of our **favorite complexity classes**.

Consequences of Isomorphism and Gap Theorems

For nice complexity classes, all AC^0 complete sets are AC^0 isomorphic.

- **Morally**, the **BH Isomorphism Conjecture** is **true**.
- Each **nice complexity class** has exactly **one complete problem**.
- **Dichotomy Phenomenon: "Natural"** computational problems tend to be **complete via fops** for one of our **favorite complexity classes**.
- Great for **Algorithms** and **Complexity Theory**!

Consequences of Isomorphism and Gap Theorems

For nice complexity classes, all AC^0 complete sets are AC^0 isomorphic.

- **Morally**, the **BH Isomorphism Conjecture** is **true**.
- Each **nice complexity class** has exactly **one complete problem**.
- **Dichotomy Phenomenon: "Natural"** computational problems tend to be **complete via fops** for one of our **favorite complexity classes**.
- Great for **Algorithms** and **Complexity Theory**!
- But not true in general [L75].

Consequences of Isomorphism and Gap Theorems

For nice complexity classes, all AC^0 complete sets are AC^0 isomorphic.

- **Morally**, the **BH Isomorphism Conjecture** is **true**.
- Each **nice complexity class** has exactly **one complete problem**.
- **Dichotomy Phenomenon: "Natural"** computational problems tend to be **complete via fops** for one of our **favorite complexity classes**.
- Great for **Algorithms** and **Complexity Theory**!
- But not true in general [L75].
- Why does this seem to occur?

Consequences of Isomorphism and Gap Theorems

For nice complexity classes, all AC^0 complete sets are AC^0 isomorphic.

- **Morally**, the **BH Isomorphism Conjecture** is **true**.
- Each **nice complexity class** has exactly **one complete problem**.
- **Dichotomy Phenomenon: "Natural"** computational problems tend to be **complete via fops** for one of our **favorite complexity classes**.
- Great for **Algorithms** and **Complexity Theory**!
- But not true in general [L75].
- Why does this seem to occur?
- **Logical** and **Algebraic** reasons, e.g., CSP.

Consequences of Isomorphism and Gap Theorems

For nice complexity classes, all AC^0 complete sets are AC^0 isomorphic.

- **Morally**, the **BH Isomorphism Conjecture** is **true**.
- Each **nice complexity class** has exactly **one complete problem**.
- **Dichotomy Phenomenon: "Natural"** computational problems tend to be **complete via fops** for one of our **favorite complexity classes**.
- Great for **Algorithms** and **Complexity Theory**!
- But not true in general [L75].
- Why does this seem to occur?
- **Logical** and **Algebraic** reasons, e.g., CSP.
- Can we remove the non-uniformity?

Consequences of Isomorphism and Gap Theorems

For nice complexity classes, all AC^0 complete sets are AC^0 isomorphic.

- **Morally**, the **BH Isomorphism Conjecture** is **true**.
- Each **nice complexity class** has exactly **one complete problem**.
- **Dichotomy Phenomenon: "Natural"** computational problems tend to be **complete via fops** for one of our **favorite complexity classes**.
- Great for **Algorithms** and **Complexity Theory**!
- But not true in general [L75].
- Why does this seem to occur?
- **Logical** and **Algebraic** reasons, e.g., CSP.
- Can we remove the non-uniformity?
- Yes! [Ag01] "The First-Order Isomorphism Theorem"

Thank you, Michal and Martin!

Thank you, Michal and Martin!

Thank you and Congratulations, Eric and Mike!

Thank you, Michal and Martin!

Thank you and Congratulations, Eric and Mike!

Enjoy the brunch tomorrow!

Thank you, Michal and Martin!

Thank you and Congratulations, Eric and Mike!

Enjoy the brunch tomorrow!

Don’t shy away too much from hard problems, ...

Thank you, Michal and Martin!

Thank you and Congratulations, Eric and Mike!

Enjoy the brunch tomorrow!

Don’t shy away too much from hard problems, . . .

. . . , especially after you have tenure.