Near-Optimal Lower Bounds on Quantifier Depth and Weisfeiler–Leman Refinement Steps

Jakob Nordström

KTH Royal Institute of Technology
Stockholm, Sweden

Datalogisk Institut på Københavns Universitet
September 6, 2018

Joint work with Christoph Berkholz

k-Variable Fragments of First-Order Logic

Two vertices are connected by a path of length 4:

$$\varphi_{\textsf{dist-4}}(x, y) = \exists z_1 \exists z_2 \exists z_3 \left(Exz_1 \wedge Ez_1z_2 \wedge Ez_2z_3 \wedge Ez_3y \right)$$

k-Variable Fragments of First-Order Logic

Two vertices are connected by a path of length 4:

$$\varphi_{\textsf{dist-4}}(x,y) = \exists z_1 \exists z_2 \exists z_3 \left(Exz_1 \land Ez_1z_2 \land Ez_2z_3 \land Ez_3y \right)$$

Equivalent $\mathcal{L}^3$ formula:

$$\varphi'_{\textsf{dist-4}}(x,y) = \exists z \Big(Exz \land \exists x \left(Ezx \land \exists z \left(Exz \land Ezy \right) \right) \Big)$$

k-Variable Fragments of First-Order Logic

Two vertices are connected by a path of length 4:

$$\varphi_{\textsf{dist-4}}(x, y) = \exists z_1 \exists z_2 \exists z_3 \left(Exz_1 \land Ez_1z_2 \land Ez_2z_3 \land Ez_3y \right)$$

Equivalent $\mathcal{L}^3$ formula:

$$\varphi'_{\textsf{dist-4}}(x, y) = \exists z \Big(Exz \land \exists x \left(Ezx \land \exists z \left(Exz \land Ezy \right) \right) \Big)$$

$\mathcal{C}^k$ extends $\mathcal{L}^k$ by counting quantifiers $\exists^{\geq i} x$

k-Variable Fragments of First-Order Logic

Two vertices are connected by a path of length 4:

$$\varphi_{\textsf{dist-4}}(x,y) = \exists z_1 \exists z_2 \exists z_3 \left(Exz_1 \wedge Ez_1 z_2 \wedge Ez_2 z_3 \wedge Ez_3 y \right)$$

Equivalent $\mathcal{L}^3$ formula:

$$\varphi'_{\textsf{dist-4}}(x,y) = \exists z \Big(Exz \wedge \exists x \left(Ezx \wedge \exists z \left(Exz \wedge Ezy \right) \right) \Big)$$

$\mathcal{C}^k$ extends $\mathcal{L}^k$ by counting quantifiers $\exists^{\geq i} x$

Vertex has degree ≥ 7:

$$\varphi_{\textsf{deg-7}}(x) = \exists y_1 \cdots \exists y_7 \bigwedge_{i \neq j} y_i \neq y_j \bigwedge_i Exy_i$$

k-Variable Fragments of First-Order Logic

Two vertices are connected by a path of length 4:

$$\varphi_{\textsf{dist-4}}(x, y) = \exists z_1 \exists z_2 \exists z_3 \left(Exz_1 \wedge Ez_1z_2 \wedge Ez_2z_3 \wedge Ez_3y \right)$$

Equivalent $\mathcal{L}^3$ formula:

$$\varphi'_{\textsf{dist-4}}(x, y) = \exists z \Big(Exz \wedge \exists x \left(Ezx \wedge \exists z \left(Exz \wedge Ezy \right)\right)\Big)$$

$\mathcal{C}^k$ extends $\mathcal{L}^k$ by counting quantifiers $\exists^{\geq i} x$

Vertex has degree ≥ 7:

$$\varphi_{\textsf{deg-7}}(x) = \exists y_1 \cdots \exists y_7 \bigwedge_{i \neq j} y_i \neq y_j \bigwedge_i Exy_i$$

Equivalent $\mathcal{C}^2$ formula:

$$\varphi'_{\textsf{deg-7}}(x) = \exists^{\geq 7} y \, Exy$$

Finite Relational Structures

- Structure $\mathcal{A}$
- Domain $V(\mathcal{A}) = \{u_1, u_2, \ldots, u_n\}$
- Relations R_ℓ of arity r_ℓ
- Interpretation $R_\ell^{\mathcal{A}} = \left\{(u_{j_1}, \ldots, u_{j_\ell}) \middle| \text{relation } R_\ell u_{j_1}, \ldots, u_{j_\ell} \text{ holds}\right\}$
- $\mathcal{A} \models \varphi$ if sentence φ true in structure $\mathcal{A}$
- Running example: graphs
  - Elements: vertices
  - Relations: edges

Why Bounded Variable Fragments of First Order Logic?

Numerous applications in finite model theory and related areas [Gro98]

Why Bounded Variable Fragments of First Order Logic?

Numerous applications in finite model theory and related areas [Gro98]

Model checking problem

Given finite relational structure $\mathcal{A}$ and sentence φ, does $\mathcal{A}$ satisfy φ?

Decidable in polynomial time [Imm82, Var95]

Why Bounded Variable Fragments of First Order Logic?

Numerous applications in finite model theory and related areas [Gro98]

Model checking problem

Given finite relational structure $\mathcal{A}$ and sentence φ, does $\mathcal{A}$ satisfy φ?

Decidable in polynomial time [Imm82, Var95]

Equivalence problem

Given two finite relational structures $\mathcal{A}$ and $\mathcal{B}$, do they satisfy the same $\mathcal{L}^k$ or $\mathcal{C}^k$ sentences?

Decidable in time $n^{O(k)}$ [IL90] (i.e., polynomial for constant k)

Connections to Weisfeiler–Leman

- Equivalence problem for $\mathcal{C}^{k+1}$ closely related to k-dimensional Weisfeiler–Leman algorithm (k-WL) for testing non-isomorphism of
  - graphs
  - more general relational structures

- $\mathcal{A}$ and $\mathcal{B}$ distinguished by k-dimensional Weisfeiler–Leman $\Leftrightarrow$ $\exists\ \mathcal{C}^{k+1}$ sentence differentiating between $\mathcal{A}$ and $\mathcal{B}$ [CFI92]

- Quantifier depth of distinguishing $\mathcal{C}^{k+1}$ sentence $=$
  $=$ #iterations k-WL needs to tell $\mathcal{A}$ and $\mathcal{B}$ apart

The Weisfeiler–Leman Algorithm

- Introduced by Babai in 1979 and Immerman and Lander [IL90]
- Iteratively refines colouring of element set
- Ends with canonical stable colouring classifying similar elements
- For parameter k, runs in time $n^{O(k)}$
- Reduces search space (isomorphisms preserve similar elements)
- In particular: different stable colourings $\Rightarrow$ non-isomorphic structures

The Weisfeiler–Leman Algorithm

- Introduced by Babai in 1979 and Immerman and Lander [IL90]
- Iteratively refines colouring of element set
- Ends with canonical stable colouring classifying similar elements
- For parameter k, runs in time $n^{O(k)}$
- Reduces search space (isomorphisms preserve similar elements)
- In particular: different stable colourings $\Rightarrow$ non-isomorphic structures

Graph isomorphism for minor-free graphs [Gro12]

For every nontrivial graph class excluding some minor (e.g., planar graphs; graphs of bounded treewidth) $\exists\ k$ such that k-WL decides isomorphism

The Weisfeiler–Leman Algorithm

- Introduced by Babai in 1979 and Immerman and Lander [IL90]
- Iteratively refines colouring of element set
- Ends with canonical stable colouring classifying similar elements
- For parameter k, runs in time $n^{O(k)}$
- Reduces search space (isomorphisms preserve similar elements)
- In particular: different stable colourings $\Rightarrow$ non-isomorphic structures

Graph isomorphism for minor-free graphs [Gro12]

For every nontrivial graph class excluding some minor (e.g., planar graphs; graphs of bounded treewidth) $\exists\ k$ such that k-WL decides isomorphism

Babai's general graph isomorphism algorithm [Bab16]

Applies k-dimensional Weisfeiler–Leman for polylogarithmic k
$\Rightarrow$ quasipolynomial running time

Quantifier Depth of $\mathcal{C}^k$

Definition

$\mathrm{D}^k(\mathcal{A}, \mathcal{B})$: minimal quantifier depth of $\mathcal{C}^k$ sentence distinguishing two n-element structures $\mathcal{A}$ and $\mathcal{B}$ (with $\mathcal{A} \not\equiv_{\mathcal{C}^k} \mathcal{B}$)

Quantifier Depth of $\mathcal{C}^k$

Definition

$\mathrm{D}^k(\mathcal{A},\mathcal{B})$: minimal quantifier depth of $\mathcal{C}^k$ sentence distinguishing two n-element structures $\mathcal{A}$ and $\mathcal{B}$ (with $\mathcal{A} \not\equiv_{\mathcal{C}^k} \mathcal{B}$)

- $\mathrm{D}^n(\mathcal{A},\mathcal{B}) \leq n$

$$\exists x_1 \cdots \exists x_n \left(\bigwedge_{i \neq j} x_i \neq x_j \ \wedge \bigwedge_{\substack{R \in \sigma, \\ (v_{i_1},\ldots,v_{i_r}) \in R^{\mathcal{A}}}} R x_{i_1} \cdots x_{i_r} \ \wedge \bigwedge_{\substack{R \in \sigma, \\ (v_{i_1},\ldots,v_{i_r}) \notin R^{\mathcal{A}}}} \neg R x_{i_1} \cdots x_{i_r} \right)$$

Quantifier Depth of $\mathcal{C}^k$

Definition

$\mathrm{D}^k(\mathcal{A},\mathcal{B})$: minimal quantifier depth of $\mathcal{C}^k$ sentence distinguishing two n-element structures $\mathcal{A}$ and $\mathcal{B}$ (with $\mathcal{A} \not\equiv_{\mathcal{C}^k} \mathcal{B}$)

- $\mathrm{D}^n(\mathcal{A},\mathcal{B}) \leq n$

$$\exists x_1 \cdots \exists x_n \left(\bigwedge_{i \neq j} x_i \neq x_j \;\wedge \bigwedge_{\substack{R \in \sigma, \\ (v_{i_1},\ldots,v_{i_r}) \in R^{\mathcal{A}}}} R x_{i_1} \cdots x_{i_r} \;\wedge \bigwedge_{\substack{R \in \sigma, \\ (v_{i_1},\ldots,v_{i_r}) \notin R^{\mathcal{A}}}} \neg R x_{i_1} \cdots x_{i_r} \right)$$

- $\mathrm{D}^k(\mathcal{A},\mathcal{B}) \leq n^{k-1}$

Quantifier Depth of $\mathcal{C}^k$

Definition

$\mathrm{D}^k(\mathcal{A},\mathcal{B})$: minimal quantifier depth of $\mathcal{C}^k$ sentence distinguishing two n-element structures $\mathcal{A}$ and $\mathcal{B}$ (with $\mathcal{A} \not\equiv_{\mathcal{C}^k} \mathcal{B}$)

- $\mathrm{D}^n(\mathcal{A},\mathcal{B}) \leq n$

$$\exists x_1 \cdots \exists x_n \left(\bigwedge_{i \neq j} x_i \neq x_j \;\wedge \bigwedge_{\substack{R \in \sigma, \\ (v_{i_1},\ldots,v_{i_r}) \in R^{\mathcal{A}}}} R x_{i_1} \cdots x_{i_r} \;\wedge \bigwedge_{\substack{R \in \sigma, \\ (v_{i_1},\ldots,v_{i_r}) \notin R^{\mathcal{A}}}} \neg R x_{i_1} \cdots x_{i_r} \right)$$

- $\mathrm{D}^k(\mathcal{A},\mathcal{B}) \leq n^{k-1}$

$\mathrm{D}^3(\mathcal{A},\mathcal{B}) \leq \mathcal{O}\left(n^2 / \log n\right)$ [KS16]

Quantifier Depth of $\mathcal{C}^k$

Definition

$\mathrm{D}^k(\mathcal{A},\mathcal{B})$: minimal quantifier depth of $\mathcal{C}^k$ sentence distinguishing two n-element structures $\mathcal{A}$ and $\mathcal{B}$ (with $\mathcal{A} \not\equiv_{\mathcal{C}^k} \mathcal{B}$)

- $\mathrm{D}^n(\mathcal{A},\mathcal{B}) \leq n$

$$\exists x_1 \cdots \exists x_n \left(\bigwedge_{i \neq j} x_i \neq x_j \;\wedge \bigwedge_{\substack{R \in \sigma, \\ (v_{i_1},\ldots,v_{i_r}) \in R^{\mathcal{A}}}} R x_{i_1} \cdots x_{i_r} \;\wedge \bigwedge_{\substack{R \in \sigma, \\ (v_{i_1},\ldots,v_{i_r}) \notin R^{\mathcal{A}}}} \neg R x_{i_1} \cdots x_{i_r} \right)$$

- $\mathrm{D}^k(\mathcal{A},\mathcal{B}) \leq n^{k-1}$ $\qquad \mathrm{D}^3(\mathcal{A},\mathcal{B}) \leq \mathcal{O}\big(n^2/\log n\big)$ [KS16]
- k constant: $\mathrm{D}^k(\mathcal{A},\mathcal{B}) \geq \Omega\big(n\big)$ [Gro99, Für01, KV15]

Quantifier Depth of $\mathcal{C}^k$

Definition

$\mathrm{D}^k(\mathcal{A},\mathcal{B})$: minimal quantifier depth of $\mathcal{C}^k$ sentence distinguishing two n-element structures $\mathcal{A}$ and $\mathcal{B}$ (with $\mathcal{A} \not\equiv_{\mathcal{C}^k} \mathcal{B}$)

- $\mathrm{D}^n(\mathcal{A},\mathcal{B}) \leq n$

$$\exists x_1 \cdots \exists x_n \left(\bigwedge_{i \neq j} x_i \neq x_j \;\wedge \bigwedge_{\substack{R \in \sigma, \\ (v_{i_1},\ldots,v_{i_r}) \in R^{\mathcal{A}}}} R x_{i_1} \cdots x_{i_r} \;\wedge \bigwedge_{\substack{R \in \sigma, \\ (v_{i_1},\ldots,v_{i_r}) \notin R^{\mathcal{A}}}} \neg R x_{i_1} \cdots x_{i_r} \right)$$

- $\mathrm{D}^k(\mathcal{A},\mathcal{B}) \leq n^{k-1}$ $\qquad$ $\mathrm{D}^3(\mathcal{A},\mathcal{B}) \leq \mathcal{O}\big(n^2/\log n\big)$ [KS16]
- k constant: $\mathrm{D}^k(\mathcal{A},\mathcal{B}) \geq \Omega\big(n\big)$ [Gro99, Für01, KV15]

Theorem [BN16a]

For every $k \leq n^{0.01}$ there are n-element relational structures $\mathcal{A}$, $\mathcal{B}$ of arity $k-1$ such that $\mathrm{D}^k(\mathcal{A},\mathcal{B}) \geq n^{\Omega(k/\log k)}$

$\mathcal{C}^k$ and Weisfeiler–Leman

Theorem [BN16a]

For every $k \leq n^{0.01}$ there are n-element relational structures $\mathcal{A}$, $\mathcal{B}$ of arity $k-1$ such that $\mathrm{D}^k(\mathcal{A}, \mathcal{B}) \geq n^{\Omega(k/\log k)}$

$\mathcal{C}^k$ and Weisfeiler–Leman

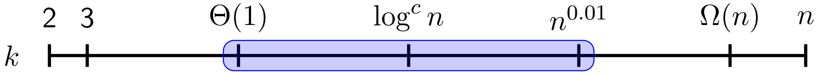
Theorem [BN16a]

For every $k \leq n^{0.01}$ there are n-element relational structures $\mathcal{A}$, $\mathcal{B}$ of arity $k-1$ such that $\mathrm{D}^k(\mathcal{A},\mathcal{B}) \geq n^{\Omega(k/\log k)}$

Theorem [CFI92]

$\mathrm{D}^k(\mathcal{A},\mathcal{B}) = $ #refinement steps $(k-1)$-dimensional Weisfeiler–Leman needs to distinguish $\mathcal{A}$ and $\mathcal{B}$

$\mathcal{C}^k$ and Weisfeiler–Leman



Theorem [BN16a]

For every $k \leq n^{0.01}$ there are n-element relational structures $\mathcal{A}$, $\mathcal{B}$ of arity $k-1$ such that $\mathrm{D}^k(\mathcal{A},\mathcal{B}) \geq n^{\Omega(k/\log k)}$

Theorem [CFI92]

$\mathrm{D}^k(\mathcal{A},\mathcal{B}) =$ #refinement steps $(k-1)$-dimensional Weisfeiler–Leman needs to distinguish $\mathcal{A}$ and $\mathcal{B}$

Application for non-constant k

- Babai's quasipolynomial graph isomorphism test uses $k = \log^c n$ on $(k-1)$-ary relational structures [Bab16]
- Our result implies $\Omega\left(n^{\log^{c-1} n}\right)$ lower bound in this setting

Overview of proof

Essence of Proof

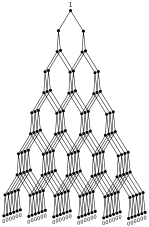
In one sentence, a novel combination of methods from

Descriptive complexity **Proof complexity**

Essence of Proof

In one sentence, a novel combination of methods from

Descriptive complexity



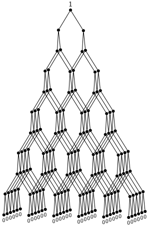
pyramid construction
Immerman [Imm81]

Proof complexity

Essence of Proof

In one sentence, a novel combination of methods from

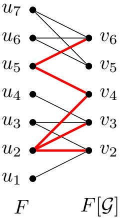
Descriptive complexity



pyramid construction
Immerman [Imm81]

$+$

Proof complexity

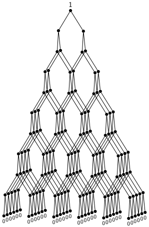


hardness condensation
Razborov [Raz16a]

Essence of Proof

In one sentence, a novel combination of methods from

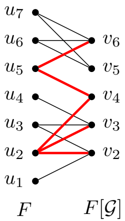
Descriptive complexity



pyramid construction
Immerman [Imm81]

$\bigoplus$

Proof complexity



hardness condensation
Razborov [Raz16a]

Connection made via XOR formulas as source of hard instances

Spoiler-Duplicator Game for $\mathcal{L}^k$

- Spoiler and Duplicator play on structures $\mathcal{A}$ and $\mathcal{B}$

Spoiler-Duplicator Game for $\mathcal{L}^k$

- Spoiler and Duplicator play on structures $\mathcal{A}$ and $\mathcal{B}$
- Positions: partial mappings $p = \{(u_1, v_1), \ldots, (u_i, v_i)\}$ from $V(\mathcal{A})$ to $V(\mathcal{B})$ of size $\leq k$ (start with empty mapping)

Spoiler-Duplicator Game for $\mathcal{L}^k$

- Spoiler and Duplicator play on structures $\mathcal{A}$ and $\mathcal{B}$
- Positions: partial mappings $p = \{(u_1, v_1), \ldots, (u_i, v_i)\}$ from $V(\mathcal{A})$ to $V(\mathcal{B})$ of size $\leq k$ (start with empty mapping)
- In each round:
  1. Spoiler chooses $p' \subseteq p$ with $|p'| < k$

Spoiler-Duplicator Game for $\mathcal{L}^k$

- Spoiler and Duplicator play on structures $\mathcal{A}$ and $\mathcal{B}$
- Positions: partial mappings $p = \{(u_1, v_1), \ldots, (u_i, v_i)\}$ from $V(\mathcal{A})$ to $V(\mathcal{B})$ of size $\leq k$ (start with empty mapping)
- In each round:
  1. Spoiler chooses $p' \subseteq p$ with $|p'| < k$
  2. Spoiler selects $u \in V(\mathcal{A})$ or $v \in V(\mathcal{B})$

Spoiler-Duplicator Game for $\mathcal{L}^k$

- Spoiler and Duplicator play on structures $\mathcal{A}$ and $\mathcal{B}$
- Positions: partial mappings $p = \{(u_1, v_1), \ldots, (u_i, v_i)\}$ from $V(\mathcal{A})$ to $V(\mathcal{B})$ of size $\leq k$ (start with empty mapping)
- In each round:
  1. Spoiler chooses $p' \subseteq p$ with $|p'| < k$
  2. Spoiler selects $u \in V(\mathcal{A})$ or $v \in V(\mathcal{B})$
  3. Duplicator responds by choosing element v or u in other structure

Spoiler-Duplicator Game for $\mathcal{L}^k$

- Spoiler and Duplicator play on structures $\mathcal{A}$ and $\mathcal{B}$
- Positions: partial mappings $p = \{(u_1, v_1), \ldots, (u_i, v_i)\}$ from $V(\mathcal{A})$ to $V(\mathcal{B})$ of size $\leq k$ (start with empty mapping)
- In each round:
  1. Spoiler chooses $p' \subseteq p$ with $|p'| < k$
  2. Spoiler selects $u \in V(\mathcal{A})$ or $v \in V(\mathcal{B})$
  3. Duplicator responds by choosing element v or u in other structure
  4. New position is $p' \cup \{(u, v)\}$

Spoiler-Duplicator Game for $\mathcal{L}^k$

- Spoiler and Duplicator play on structures $\mathcal{A}$ and $\mathcal{B}$
- Positions: partial mappings $p = \{(u_1, v_1), \ldots, (u_i, v_i)\}$ from $V(\mathcal{A})$ to $V(\mathcal{B})$ of size $\leq k$ (start with empty mapping)
- In each round:
  1. Spoiler chooses $p' \subseteq p$ with $|p'| < k$
  2. Spoiler selects $u \in V(\mathcal{A})$ or $v \in V(\mathcal{B})$
  3. Duplicator responds by choosing element v or u in other structure
  4. New position is $p' \cup \{(u, v)\}$
- Spoiler winning position: p isn't isomorphism on induced substructures

Spoiler-Duplicator Game for $\mathcal{L}^k$

- Spoiler and Duplicator play on structures $\mathcal{A}$ and $\mathcal{B}$
- Positions: partial mappings $p = \{(u_1, v_1), \ldots, (u_i, v_i)\}$ from $V(\mathcal{A})$ to $V(\mathcal{B})$ of size $\leq k$ (start with empty mapping)
- In each round:
  1. Spoiler chooses $p' \subseteq p$ with $|p'| < k$
  2. Spoiler selects $u \in V(\mathcal{A})$ or $v \in V(\mathcal{B})$
  3. Duplicator responds by choosing element v or u in other structure
  4. New position is $p' \cup \{(u, v)\}$
- Spoiler winning position: p isn't isomorphism on induced substructures

Characterization of $\mathcal{L}^k$ [Bar77, Imm82]

Spoiler wins this game for size-k mappings in R rounds $\Leftrightarrow$
$\exists$ sentence $\varphi \in \mathcal{L}^k$ of quantifier depth R such that $\mathcal{A} \models \varphi$ and $\mathcal{B} \not\models \varphi$

Spoiler-Duplicator Game for $\mathcal{C}^k$

- Spoiler and Duplicator play on structures $\mathcal{A}$ and $\mathcal{B}$
- Positions: partial mappings $p = \{(u_1, v_1), \ldots, (u_i, v_i)\}$ from $V(\mathcal{A})$ to $V(\mathcal{B})$ of size $\leq k$ (start with empty mapping)
- In each round:
    1. Spoiler chooses $p' \subseteq p$ with $|p'| < k$
    2. Duplicator selects global bijection $f : V(\mathcal{A}) \to V(\mathcal{B})$
    3. Spoiler chooses pair $(u, v) \in f$
    4. New position is $p' \cup \{(u, v)\}$
- Spoiler winning position: p isn't isomorphism on induced substructures

Spoiler-Duplicator Game for $\mathcal{C}^k$

- Spoiler and Duplicator play on structures $\mathcal{A}$ and $\mathcal{B}$
- Positions: partial mappings $p = \{(u_1, v_1), \ldots, (u_i, v_i)\}$ from $V(\mathcal{A})$ to $V(\mathcal{B})$ of size $\leq k$ (start with empty mapping)
- In each round:
  1. Spoiler chooses $p' \subseteq p$ with $|p'| < k$
  2. Duplicator selects global bijection $f : V(\mathcal{A}) \to V(\mathcal{B})$
  3. Spoiler chooses pair $(u, v) \in f$
  4. New position is $p' \cup \{(u, v)\}$
- Spoiler winning position: p isn't isomorphism on induced substructures

Characterization of $\mathcal{C}^k$ [CFI92, Hel96]

Spoiler wins this game for size-k mappings in R rounds $\Leftrightarrow$
$\exists$ sentence $\varphi \in \mathcal{C}^k$ of quantifier depth R such that $\mathcal{A} \models \varphi$ and $\mathcal{B} \not\models \varphi$

XOR Formulas

s-XOR formula F over Boolean variables $x_1, \ldots, x_n$:
set of parity constraints $x_{i_1} \oplus \cdots \oplus x_{i_r} = a$, $r \leq s$, $a \in \{0, 1\}$

XOR Formulas

s-XOR formula F over Boolean variables $x_1, \ldots, x_n$:
set of parity constraints $x_{i_1} \oplus \cdots \oplus x_{i_r} = a$, $r \leq s$, $a \in \{0, 1\}$

Let $\mathcal{A}(F)$ and $\mathcal{B}(F)$ relational structures with

- 2 vertices x_i^0, x_i^1 for every $x_i \in \mathrm{Vars}(F)$
- relations

$$X_i^{\mathcal{A}} = X_i^{\mathcal{B}} = \{x_i^0, x_i^1\}$$
$$R_r^{\mathcal{A}} = \left\{(x_{i_1}^{a_1}, \ldots, x_{i_r}^{a_r}) \middle| (x_{i_1} \oplus \cdots \oplus x_{i_r} = a) \in F,\ \bigoplus_i a_i = 0\right\}$$
$$R_r^{\mathcal{B}} = \left\{(x_{i_1}^{a_1}, \ldots, x_{i_r}^{a_r}) \middle| (x_{i_1} \oplus \cdots \oplus x_{i_r} = a) \in F,\ \bigoplus_i a_i = a\right\}$$

XOR Formulas

s-XOR formula F over Boolean variables $x_1, \ldots, x_n$:
set of parity constraints $x_{i_1} \oplus \cdots \oplus x_{i_r} = a$, $r \leq s$, $a \in \{0, 1\}$

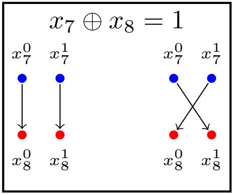
Let $\mathcal{A}(F)$ and $\mathcal{B}(F)$ relational structures with

- 2 vertices x_i^0, x_i^1 for every $x_i \in \mathrm{Vars}(F)$
- relations

$$X_i^{\mathcal{A}} = X_i^{\mathcal{B}} = \{x_i^0, x_i^1\}$$
$$R_r^{\mathcal{A}} = \left\{(x_{i_1}^{a_1}, \ldots, x_{i_r}^{a_r}) \middle| (x_{i_1} \oplus \cdots \oplus x_{i_r} = a) \in F,\ \bigoplus_i a_i = 0\right\}$$
$$R_r^{\mathcal{B}} = \left\{(x_{i_1}^{a_1}, \ldots, x_{i_r}^{a_r}) \middle| (x_{i_1} \oplus \cdots \oplus x_{i_r} = a) \in F,\ \bigoplus_i a_i = a\right\}$$

Isomorphism $I : \mathcal{A}(F) \to \mathcal{B}(F)$ corresponds to satisfying assignment α for F via

$$\alpha(x_i) = 0 \iff I(x_i^0) = x_i^0 \Leftrightarrow I(x_i^1) = x_i^1$$
$$\alpha(x_i) = 1 \iff I(x_i^0) = x_i^1 \Leftrightarrow I(x_i^1) = x_i^0$$



A Pebble Game on XOR Formulas

The k-pebble game on XOR formula F is played by two players

- Positions: partial assignments α, $|\alpha| \leq k$
- Starting position $\alpha_0 = \emptyset$

A Pebble Game on XOR Formulas

The k-pebble game on XOR formula F is played by two players

- Positions: partial assignments α, $|\alpha| \leq k$
- Starting position $\alpha_0 = \emptyset$

In round i starting from α_{i-1}:

- Player 1 chooses $\alpha \subseteq \alpha_{i-1}$, $|\alpha| < k$
- Player 1 asks for value of variable x
- Player 2 answers with $a \in \{0, 1\}$
- $\alpha_i = \alpha \cup \{x \mapsto a\}$

A Pebble Game on XOR Formulas

The k-pebble game on XOR formula F is played by two players

- Positions: partial assignments α, $|\alpha| \leq k$
- Starting position $\alpha_0 = \emptyset$

In round i starting from α_{i-1}:

- Player 1 chooses $\alpha \subseteq \alpha_{i-1}$, $|\alpha| < k$
- Player 1 asks for value of variable x
- Player 2 answers with $a \in \{0, 1\}$
- $\alpha_i = \alpha \cup \{x \mapsto a\}$

Player 1 wins game in R rounds if α_R falsifies some XOR-constraint

Equivalent Characterizations of the Pebble Game

Let

- F s-XOR formula
- $R, k \in \mathbb{N}^+,\ k > s$

Equivalent Characterizations of the Pebble Game

Let

- F s-XOR formula
- $R, k \in \mathbb{N}^+,\ k > s$

Then the following statements are equivalent:

(a) Player 1 wins R-round k-pebble game on F

Equivalent Characterizations of the Pebble Game

Let

- F s-XOR formula
- $R, k \in \mathbb{N}^+, \; k > s$

Then the following statements are equivalent:

(a) Player 1 wins R-round k-pebble game on F

(b) $\exists$ k-variable sentence $\varphi \in \mathcal{L}^k$ of quantifier depth R such that $\mathcal{A}(F) \models \varphi$ and $\mathcal{B}(F) \not\models \varphi$

Equivalent Characterizations of the Pebble Game

Let

- F s-XOR formula
- $R, k \in \mathbb{N}^+,\ k > s$

Then the following statements are equivalent:

(a) Player 1 wins R-round k-pebble game on F

(b) $\exists$ k-variable sentence $\varphi \in \mathcal{L}^k$ of quantifier depth R such that $\mathcal{A}(F) \models \varphi$ and $\mathcal{B}(F) \not\models \varphi$

(c) $\exists$ k-variable sentence $\varphi \in \mathcal{C}^k$ of quantifier depth R such that $\mathcal{A}(F) \models \varphi$ and $\mathcal{B}(F) \not\models \varphi$

Equivalent Characterizations of the Pebble Game

Let

- F s-XOR formula
- $R, k \in \mathbb{N}^+,\ k > s$

Then the following statements are equivalent:

(a) Player 1 wins R-round k-pebble game on F

(b) $\exists$ k-variable sentence $\varphi \in \mathcal{L}^k$ of quantifier depth R such that $\mathcal{A}(F) \models \varphi$ and $\mathcal{B}(F) \not\models \varphi$

(c) $\exists$ k-variable sentence $\varphi \in \mathcal{C}^k$ of quantifier depth R such that $\mathcal{A}(F) \models \varphi$ and $\mathcal{B}(F) \not\models \varphi$

(d) The s-CNF-formula $\mathrm{cnf}(F)$ has a resolution refutation of
  - depth R
  - width $k - 1$ [AD08]

Outline of Proof

[Imm81]

There are $\mathcal{A}$, $\mathcal{B}$ such that $\mathrm{D}^k(\mathcal{A}, \mathcal{B}) = \Omega\left(2^{\sqrt{\log n}}\right)$ for all $k \geq 3$

Outline of Proof

[Imm81]

There are $\mathcal{A}$, $\mathcal{B}$ such that $\mathrm{D}^k(\mathcal{A},\mathcal{B}) = \Omega\big(2^{\sqrt{\log n}}\big)$ for all $k \geq 3$

Part I (pyramid construction):

For every k there are n-variable 3-XOR formulas such that Player 1

- wins 3-pebble game for $3 \leq \ell \leq k$
- needs $n^{\Omega(1/\log k)}$ rounds to win the ℓ-pebble game for $3 \leq \ell \leq k$

Outline of Proof

[Imm81]

There are $\mathcal{A}$, $\mathcal{B}$ such that $\mathrm{D}^k(\mathcal{A}, \mathcal{B}) = \Omega\big(2^{\sqrt{\log n}}\big)$ for all $k \geq 3$

Part I (pyramid construction):

For every k there are n-variable 3-XOR formulas such that Player 1

- wins 3-pebble game for $3 \leq \ell \leq k$
- needs $n^{\Omega(1/\log k)}$ rounds to win the ℓ-pebble game for $3 \leq \ell \leq k$

Part II (hardness condensation):

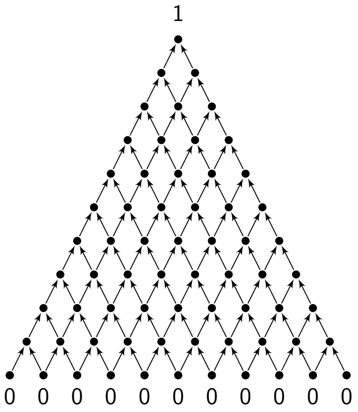
Reduce the number of variables without destroying the lower bound

Transform n-variable 3-XOR into m-variable k-XOR for $m \approx n^{1/k}$

Lower bound remains $n^{\Omega(1/\log k)} = m^{\Omega(k/\log k)}$

PART I: An $n^{\Omega(\frac{1}{\log k})}$ lower bound

A 2-Dimensional Pyramid



XORs from DAGs

Let $\mathcal{G}$ directed acyclic graph with unique sink z

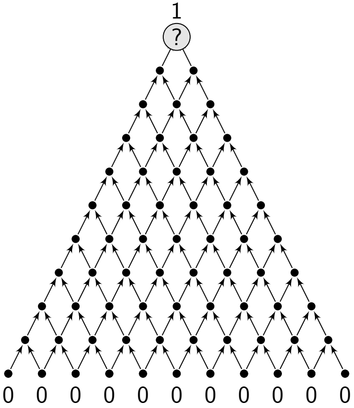
XOR-formula $\mathrm{xor}(\mathcal{G})$ over variables $v \in V(\mathcal{G})$ contains constraints:

(i) $v \oplus \bigoplus_{w \in N^-(v)} w = 0$

(ii) $s = 0$ for every source s

(iii) $z = 1$ for unique sink z

A 2-Dimensional Pyramid



XORs from DAGs

Let $\mathcal{G}$ directed acyclic graph with unique sink z

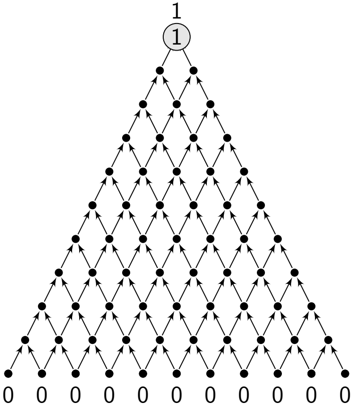
XOR-formula $\mathrm{xor}(\mathcal{G})$ over variables $v \in V(\mathcal{G})$ contains constraints:

(i) $v \oplus \bigoplus_{w \in N^-(v)} w = 0$

(ii) $s = 0$ for every source s

(iii) $z = 1$ for unique sink z

A 2-Dimensional Pyramid



XORs from DAGs

Let $\mathcal{G}$ directed acyclic graph with unique sink z

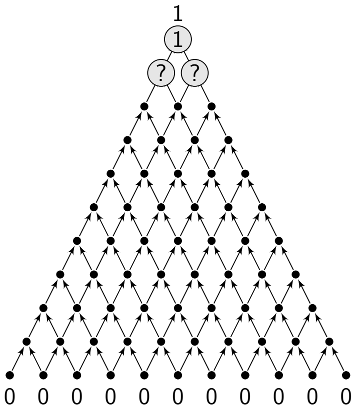
XOR-formula $\mathrm{xor}(\mathcal{G})$ over variables $v \in V(\mathcal{G})$ contains constraints:

(i) $v \oplus \bigoplus_{w \in N^-(v)} w = 0$

(ii) $s = 0$ for every source s

(iii) $z = 1$ for unique sink z

A 2-Dimensional Pyramid



XORs from DAGs

Let $\mathcal{G}$ directed acyclic graph with unique sink z

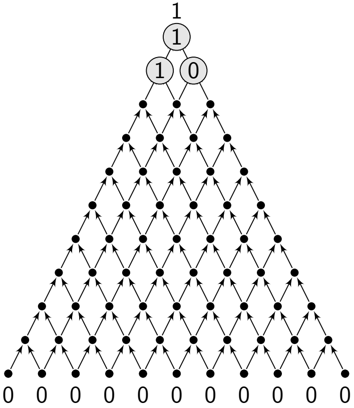
XOR-formula $\mathrm{xor}(\mathcal{G})$ over variables $v \in V(\mathcal{G})$ contains constraints:

(i) $v \oplus \bigoplus_{w \in N^-(v)} w = 0$

(ii) $s = 0$ for every source s

(iii) $z = 1$ for unique sink z

A 2-Dimensional Pyramid



XORs from DAGs

Let $\mathcal{G}$ directed acyclic graph with unique sink z

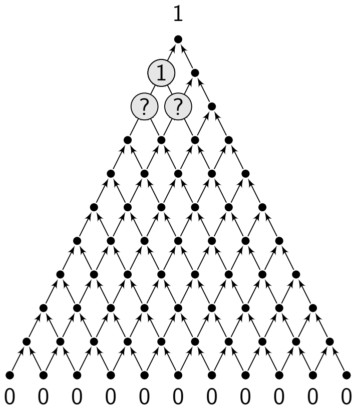
XOR-formula $\mathrm{xor}(\mathcal{G})$ over variables $v \in V(\mathcal{G})$ contains constraints:

(i) $v \oplus \bigoplus_{w \in N^-(v)} w = 0$

(ii) $s = 0$ for every source s

(iii) $z = 1$ for unique sink z

A 2-Dimensional Pyramid



XORs from DAGs

Let $\mathcal{G}$ directed acyclic graph with unique sink z

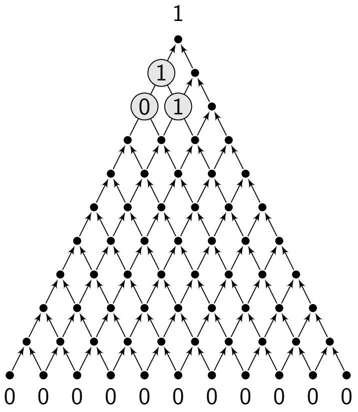
XOR-formula $\mathrm{xor}(\mathcal{G})$ over variables $v \in V(\mathcal{G})$ contains constraints:

(i) $v \oplus \bigoplus_{w \in N^-(v)} w = 0$

(ii) $s = 0$ for every source s

(iii) $z = 1$ for unique sink z

A 2-Dimensional Pyramid



XORs from DAGs

Let $\mathcal{G}$ directed acyclic graph with unique sink z

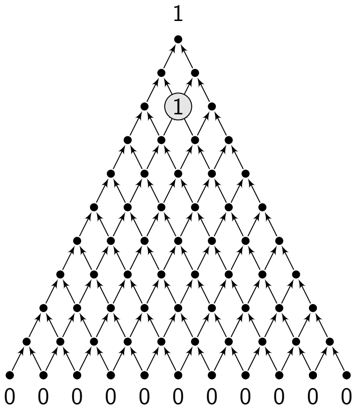
XOR-formula $\mathrm{xor}(\mathcal{G})$ over variables $v \in V(\mathcal{G})$ contains constraints:

(i) $v \oplus \bigoplus_{w \in N^-(v)} w = 0$

(ii) $s = 0$ for every source s

(iii) $z = 1$ for unique sink z

A 2-Dimensional Pyramid



XORs from DAGs

Let $\mathcal{G}$ directed acyclic graph with unique sink z

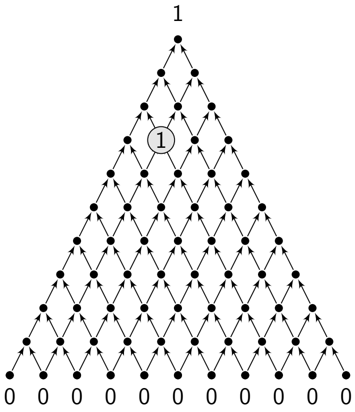
XOR-formula $\mathrm{xor}(\mathcal{G})$ over variables $v \in V(\mathcal{G})$ contains constraints:

(i) $v \oplus \bigoplus_{w \in N^-(v)} w = 0$

(ii) $s = 0$ for every source s

(iii) $z = 1$ for unique sink z

A 2-Dimensional Pyramid



XORs from DAGs

Let $\mathcal{G}$ directed acyclic graph with unique sink z

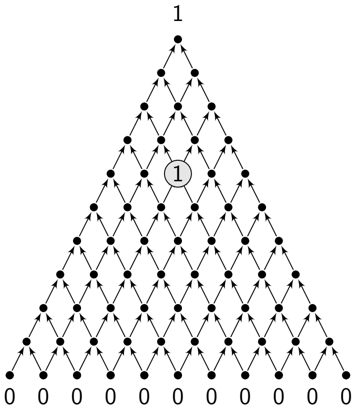
XOR-formula $\mathrm{xor}(\mathcal{G})$ over variables $v \in V(\mathcal{G})$ contains constraints:

(i) $v \oplus \bigoplus_{w \in N^-(v)} w = 0$

(ii) $s = 0$ for every source s

(iii) $z = 1$ for unique sink z

A 2-Dimensional Pyramid



XORs from DAGs

Let $\mathcal{G}$ directed acyclic graph with unique sink z

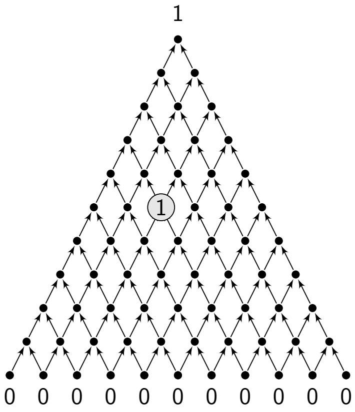
XOR-formula $\mathrm{xor}(\mathcal{G})$ over variables $v \in V(\mathcal{G})$ contains constraints:

(i) $v \oplus \bigoplus_{w \in N^-(v)} w = 0$

(ii) $s = 0$ for every source s

(iii) $z = 1$ for unique sink z

A 2-Dimensional Pyramid



XORs from DAGs

Let $\mathcal{G}$ directed acyclic graph with unique sink z

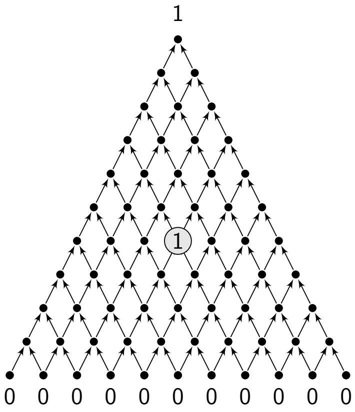
XOR-formula $\mathrm{xor}(\mathcal{G})$ over variables $v \in V(\mathcal{G})$ contains constraints:

(i) $v \oplus \bigoplus_{w \in N^-(v)} w = 0$

(ii) $s = 0$ for every source s

(iii) $z = 1$ for unique sink z

A 2-Dimensional Pyramid



XORs from DAGs

Let $\mathcal{G}$ directed acyclic graph with unique sink z

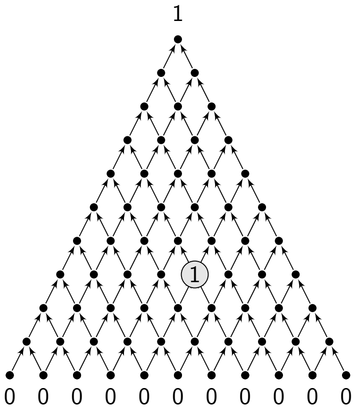
XOR-formula $\mathrm{xor}(\mathcal{G})$ over variables $v \in V(\mathcal{G})$ contains constraints:

(i) $v \oplus \bigoplus_{w \in N^{-}(v)} w = 0$

(ii) $s = 0$ for every source s

(iii) $z = 1$ for unique sink z

A 2-Dimensional Pyramid



XORs from DAGs

Let $\mathcal{G}$ directed acyclic graph with unique sink z

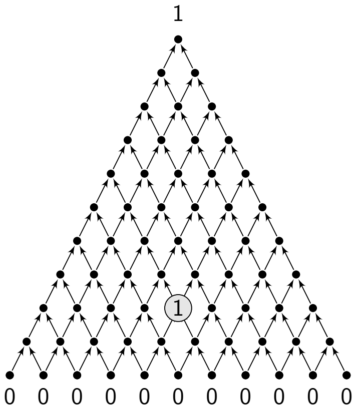
XOR-formula $\mathrm{xor}(\mathcal{G})$ over variables $v \in V(\mathcal{G})$ contains constraints:

(i) $v \oplus \bigoplus_{w \in N^-(v)} w = 0$

(ii) $s = 0$ for every source s

(iii) $z = 1$ for unique sink z

A 2-Dimensional Pyramid



XORs from DAGs

Let $\mathcal{G}$ directed acyclic graph with unique sink z

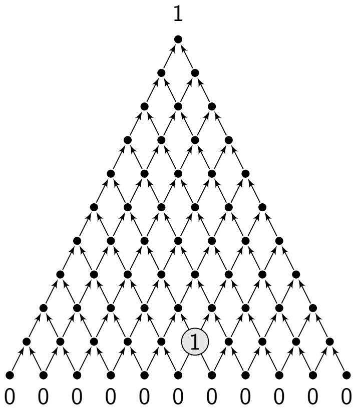
XOR-formula $\mathrm{xor}(\mathcal{G})$ over variables $v \in V(\mathcal{G})$ contains constraints:

(i) $v \oplus \bigoplus_{w \in N^-(v)} w = 0$

(ii) $s = 0$ for every source s

(iii) $z = 1$ for unique sink z

A 2-Dimensional Pyramid



XORs from DAGs

Let $\mathcal{G}$ directed acyclic graph with unique sink z

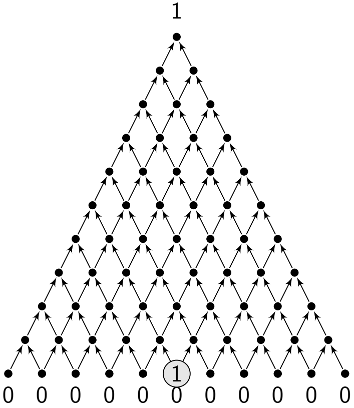
XOR-formula $\mathrm{xor}(\mathcal{G})$ over variables $v \in V(\mathcal{G})$ contains constraints:

(i) $v \oplus \bigoplus_{w \in N^-(v)} w = 0$

(ii) $s = 0$ for every source s

(iii) $z = 1$ for unique sink z

A 2-Dimensional Pyramid



XORs from DAGs

Let $\mathcal{G}$ directed acyclic graph with unique sink z

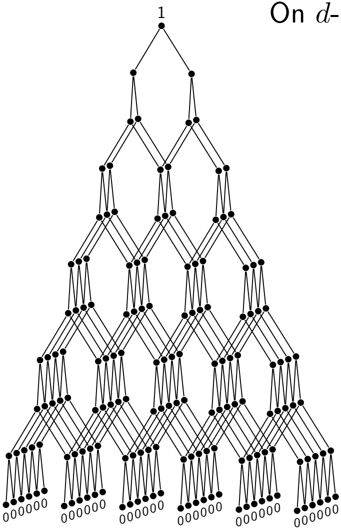
XOR-formula $\mathrm{xor}(\mathcal{G})$ over variables $v \in V(\mathcal{G})$ contains constraints:

(i) $v \oplus \bigoplus_{w \in N^-(v)} w = 0$

(ii) $s = 0$ for every source s

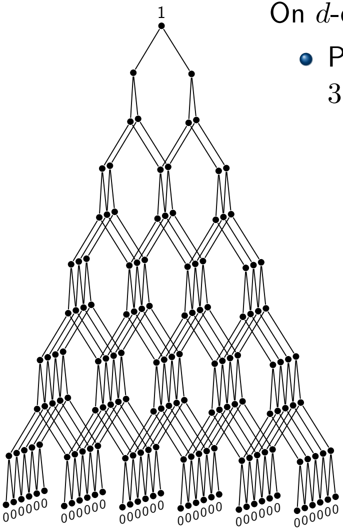
(iii) $z = 1$ for unique sink z

A 3-Dimensional Pyramid



On d-dimensional pyramid of height h

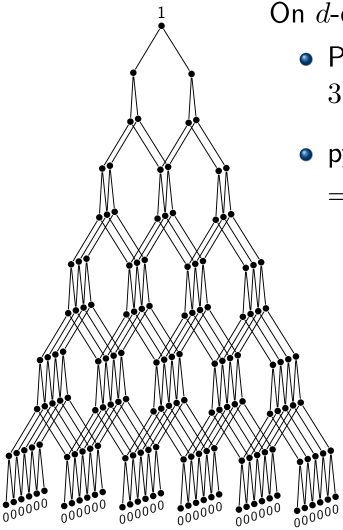
A 3-Dimensional Pyramid



On d-dimensional pyramid of height h

- Player 1 wins the k-pebble game, $3 \leq k \leq 2^{d-1}$, in $\Theta(h)$ rounds

A 3-Dimensional Pyramid



On d-dimensional pyramid of height h

- Player 1 wins the k-pebble game, $3 \leq k \leq 2^{d-1}$, in $\Theta(h)$ rounds
- pyramid has $n \approx h^d$ vertices $\Rightarrow n^{\Theta(\frac{1}{\log k})}$ rounds in k-pebble game

PART II: Hardness condensation

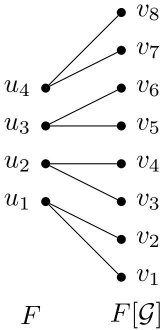
XOR Substitution with Recycling (1/2)

Suppose

- F XOR formula over variables U
- $\mathcal{G} = (U \,\dot\cup\, V, E)$ bipartite graph

Substituted formula $F[\mathcal{G}]$ over variables V:

- replace every $u \in U$ by $\bigoplus_{v \in N(u)} v$



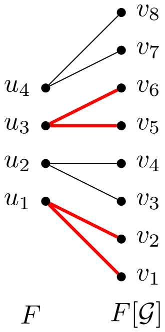
XOR Substitution with Recycling (1/2)

Suppose

- F XOR formula over variables U
- $\mathcal{G} = (U \,\dot{\cup}\, V, E)$ bipartite graph

Substituted formula $F[\mathcal{G}]$ over variables V:

- replace every $u \in U$ by $\bigoplus_{v \in N(u)} v$



$$u_1 \oplus u_3 = 1 \quad \longrightarrow \quad (v_1 \oplus v_2) \oplus (v_5 \oplus v_6) = 1$$

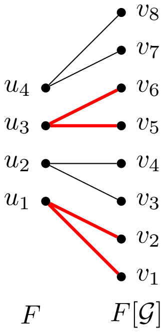
XOR Substitution with Recycling (1/2)

Suppose

- F XOR formula over variables U
- $\mathcal{G} = (U \mathbin{\dot\cup} V, E)$ bipartite graph

Substituted formula $F[\mathcal{G}]$ over variables V:

- replace every $u \in U$ by $\bigoplus_{v \in N(u)} v$



$$u_1 \oplus u_3 = 1 \quad \longrightarrow \quad (v_1 \oplus v_2) \oplus (v_5 \oplus v_6) = 1$$

Player 2 survives R-round k-pebble game on F
$\Rightarrow$ survives $2R$-round $2k$-pebble game on $F[\mathcal{G}]$

But #variables in instance goes up

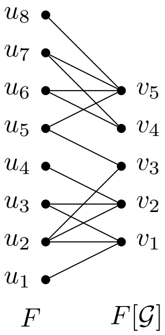
XOR Substitution with Recycling (1/2)

Suppose

- F XOR formula over variables U
- $\mathcal{G} = (U \dot{\cup} V, E)$ bipartite graph

Substituted formula $F[\mathcal{G}]$ over variables V:

- replace every $u \in U$ by $\bigoplus_{v \in N(u)} v$



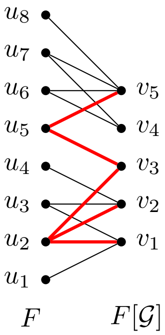
XOR Substitution with Recycling (1/2)

Suppose

- F XOR formula over variables U
- $\mathcal{G} = (U \mathbin{\dot\cup} V, E)$ bipartite graph

Substituted formula $F[\mathcal{G}]$ over variables V:

- replace every $u \in U$ by $\bigoplus_{v \in N(u)} v$

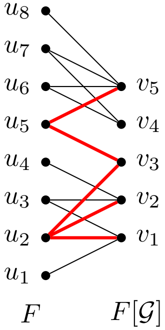


$$u_2 \oplus u_5 = 0 \quad \longrightarrow \quad (v_1 \oplus v_2 \oplus v_3) \oplus (v_3 \oplus v_5) = 0$$

Now #variables in instance goes down

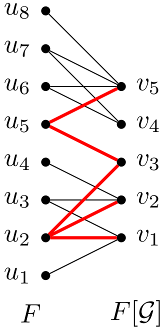
Possible to maintain hardness?

XOR Substitution with Recycling (2/2)



$$u_2 \oplus u_5 = 0 \quad \longrightarrow \quad (v_1 \oplus v_2 \oplus v_3) \oplus (v_3 \oplus v_5) = 0$$

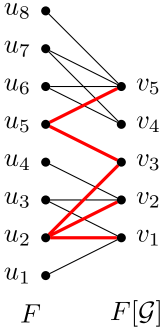
XOR Substitution with Recycling (2/2)



$$u_2 \oplus u_5 = 0 \quad \longrightarrow \quad (v_1 \oplus v_2 \oplus v_3) \oplus (v_3 \oplus v_5) = 0$$

- Apply to XOR formulas over Immerman's pyramids [Imm81]
  - Player 1 wins with 3 pebbles
  - but needs $n^{\Omega(1/\log k)}$ rounds

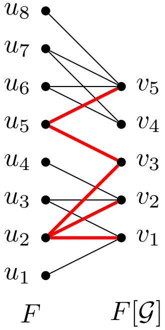
XOR Substitution with Recycling (2/2)



$$u_2 \oplus u_5 = 0 \quad \longrightarrow \quad (v_1 \oplus v_2 \oplus v_3) \oplus (v_3 \oplus v_5) = 0$$

- Apply to XOR formulas over Immerman's pyramids [Imm81]
  - Player 1 wins with 3 pebbles
  - but needs $n^{\Omega(1/\log k)}$ rounds
- $\mathcal{G}$ with left-degree $\leq k/3$, $|U| = n$, and $|V| = n^{\mathcal{O}(1/k)}$

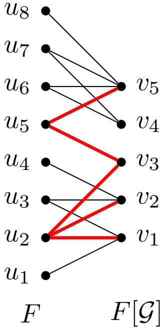
XOR Substitution with Recycling (2/2)



$$u_2 \oplus u_5 = 0 \quad \longrightarrow \quad (v_1 \oplus v_2 \oplus v_3) \oplus (v_3 \oplus v_5) = 0$$

- Apply to XOR formulas over Immerman's pyramids [Imm81]
  - Player 1 wins with 3 pebbles
  - but needs $n^{\Omega(1/\log k)}$ rounds
- $\mathcal{G}$ with left-degree $\leq k/3$, $|U| = n$, and $|V| = n^{\mathcal{O}(1/k)}$
  - Player 1 wins with k pebbles on $F[\mathcal{G}]$

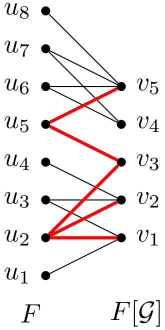
XOR Substitution with Recycling (2/2)



$$u_2 \oplus u_5 = 0 \quad \longrightarrow \quad (v_1 \oplus v_2 \oplus v_3) \oplus (v_3 \oplus v_5) = 0$$

- Apply to XOR formulas over Immerman's pyramids [Imm81]
  - Player 1 wins with 3 pebbles
  - but needs $n^{\Omega(1/\log k)}$ rounds
- $\mathcal{G}$ with left-degree $\leq k/3$, $|U| = n$, and $|V| = n^{\mathcal{O}(1/k)}$
  - Player 1 wins with k pebbles on $F[\mathcal{G}]$ ✓

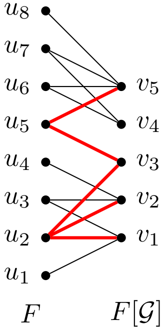
XOR Substitution with Recycling (2/2)



$$u_2 \oplus u_5 = 0 \quad \longrightarrow \quad (v_1 \oplus v_2 \oplus v_3) \oplus (v_3 \oplus v_5) = 0$$

- Apply to XOR formulas over Immerman's pyramids [Imm81]
  - Player 1 wins with 3 pebbles
  - but needs $n^{\Omega(1/\log k)}$ rounds
- $\mathcal{G}$ with left-degree $\leq k/3$, $|U| = n$, and $|V| = n^{\mathcal{O}(1/k)}$
  - Player 1 wins with k pebbles on $F[\mathcal{G}]$ ✓
  - #rounds needed for $F[\mathcal{G}] \gtrapprox$
    #rounds needed for $F = \Omega\left(|U|^{1/\log k}\right) = \Omega\left(|V|^{k/\log k}\right)$

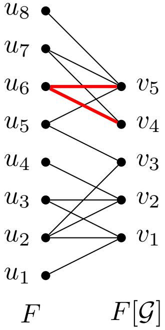
XOR Substitution with Recycling (2/2)



$$u_2 \oplus u_5 = 0 \quad \longrightarrow \quad (v_1 \oplus v_2 \oplus v_3) \oplus (v_3 \oplus v_5) = 0$$

- Apply to XOR formulas over Immerman's pyramids [Imm81]
  - Player 1 wins with 3 pebbles
  - but needs $n^{\Omega(1/\log k)}$ rounds
- $\mathcal{G}$ with left-degree $\leq k/3$, $|U| = n$, and $|V| = n^{\mathcal{O}(1/k)}$
  - Player 1 wins with k pebbles on $F[\mathcal{G}]$ ✓
  - #rounds needed for $F[\mathcal{G}] \gtrapprox$ #rounds needed for $F = \Omega\left(|U|^{1/\log k}\right) = \Omega\left(|V|^{k/\log k}\right)$?

XOR Substitution with Recycling (2/2)

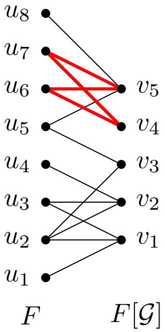


$$u_2 \oplus u_5 = 0 \quad \longrightarrow \quad (v_1 \oplus v_2 \oplus v_3) \oplus (v_3 \oplus v_5) = 0$$

$$u_6 = 1 \quad \longrightarrow \quad v_4 \oplus v_5 = 1$$

- Apply to XOR formulas over Immerman's pyramids [Imm81]
  - Player 1 wins with 3 pebbles
  - but needs $n^{\Omega(1/\log k)}$ rounds
- $\mathcal{G}$ with left-degree $\leq k/3$, $|U| = n$, and $|V| = n^{\mathcal{O}(1/k)}$
  - Player 1 wins with k pebbles on $F[\mathcal{G}]$ ✓
  - #rounds needed for $F[\mathcal{G}] \gtrapprox$
    #rounds needed for $F = \Omega\left(|U|^{1/\log k}\right) = \Omega\left(|V|^{k/\log k}\right)$?

XOR Substitution with Recycling (2/2)



$$u_2 \oplus u_5 = 0 \quad \longrightarrow \quad (v_1 \oplus v_2 \oplus v_3) \oplus (v_3 \oplus v_5) = 0$$

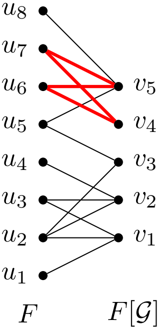
$$u_6 = 1 \quad \longrightarrow \quad v_4 \oplus v_5 = 1$$

$$u_7 = 0 \quad \longrightarrow \quad v_4 \oplus v_5 = 0$$

↯

- Apply to XOR formulas over Immerman's pyramids [Imm81]
  - Player 1 wins with 3 pebbles
  - but needs $n^{\Omega(1/\log k)}$ rounds
- $\mathcal{G}$ with left-degree $\leq k/3$, $|U| = n$, and $|V| = n^{\mathcal{O}(1/k)}$
  - Player 1 wins with k pebbles on $F[\mathcal{G}]$ ✓
  - #rounds needed for $F[\mathcal{G}] \gtrapprox$
    #rounds needed for $F = \Omega\big(|U|^{1/\log k}\big) = \Omega\big(|V|^{k/\log k}\big)$ **?**

XOR Substitution with Recycling (2/2)



$$u_2 \oplus u_5 = 0 \quad \longrightarrow \quad (v_1 \oplus v_2 \oplus v_3) \oplus (v_3 \oplus v_5) = 0$$

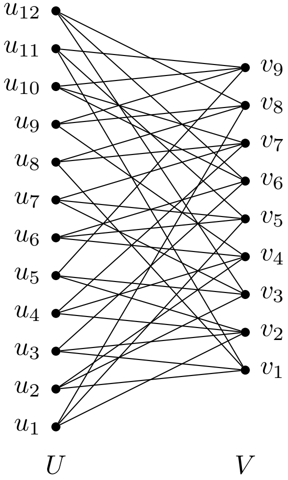
$$u_6 = 1 \quad \longrightarrow \quad v_4 \oplus v_5 = 1$$

$$u_7 = 0 \quad \longrightarrow \quad v_4 \oplus v_5 = 0$$

Solution: Use expander graphs!

- Apply to XOR formulas over Immerman's pyramids [Imm81]
  - Player 1 wins with 3 pebbles
  - but needs $n^{\Omega(1/\log k)}$ rounds
- $\mathcal{G}$ expander with left-degree $\leq k/3$, $|U| = n$, and $|V| = n^{\mathcal{O}(1/k)}$
  - Player 1 wins with k pebbles on $F[\mathcal{G}]$ ✓
  - #rounds needed for $F[\mathcal{G}] \gtrapprox$
    #rounds needed for $F = \Omega\left(|U|^{1/\log k}\right) = \Omega\left(|V|^{k/\log k}\right)$ ✓

Bipartite Boundary Expander

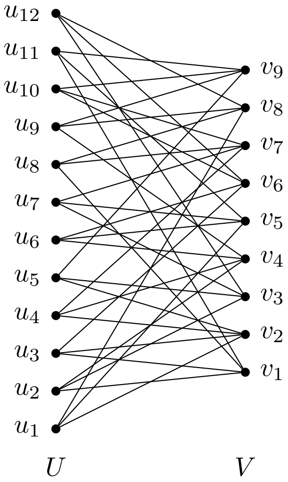


$\mathcal{G} = (U \,\dot\cup\, V, E)$ is (d, r, c)-boundary expander if

- left-degree $\leq d$
- for every $U' \subseteq U$, $|U'| \leq r$ it holds that $|\partial(U')| \geq c|U'|$

$\partial(U') = \{v \in N(U') : |N(v) \cap U'| = 1\}$

Bipartite Boundary Expander



$\mathcal{G} = (U \dot\cup V, E)$ is (d, r, c)-boundary expander if

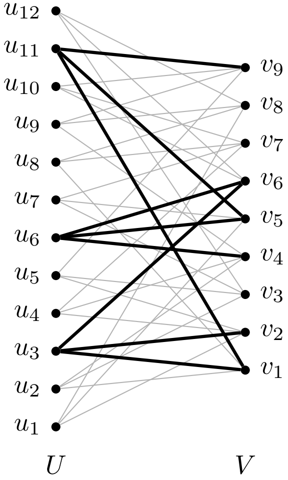
- left-degree $\leq d$
- for every $U' \subseteq U$, $|U'| \leq r$ it holds that $|\partial(U')| \geq c|U'|$

$\partial(U') = \{v \in N(U') : |N(v) \cap U'| = 1\}$

Example

- left-degree $d = 3$
- expanding set size $r = 3$
- boundary expansion factor $c = 1$

Bipartite Boundary Expander



$\mathcal{G} = (U \dot\cup V, E)$ is (d, r, c)-boundary expander if

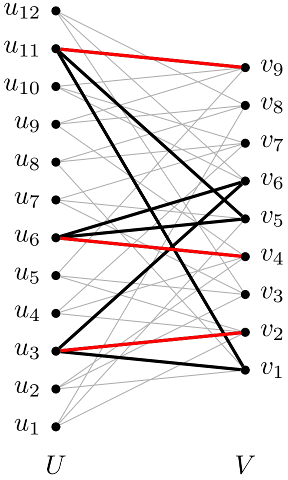
- left-degree $\leq d$
- for every $U' \subseteq U$, $|U'| \leq r$ it holds that $|\partial(U')| \geq c|U'|$

$$\partial(U') = \{v \in N(U') : |N(v) \cap U'| = 1\}$$

Example

- left-degree $d = 3$
- expanding set size $r = 3$
- boundary expansion factor $c = 1$

Bipartite Boundary Expander



$\mathcal{G} = (U \dot{\cup} V, E)$ is (d, r, c)-boundary expander if

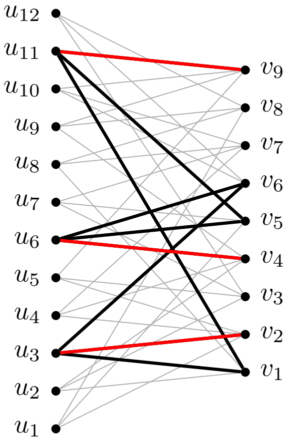
- left-degree $\leq d$
- for every $U' \subseteq U$, $|U'| \leq r$ it holds that $|\partial(U')| \geq c|U'|$

$$\partial(U') = \{v \in N(U') : |N(v) \cap U'| = 1\}$$

Example

- left-degree $d = 3$
- expanding set size $r = 3$
- boundary expansion factor $c = 1$

Bipartite Boundary Expander



$\mathcal{G} = (U \,\dot\cup\, V, E)$ is (d, r, c)-boundary expander if

- left-degree $\leq d$
- for every $U' \subseteq U$, $|U'| \leq r$ it holds that $|\partial(U')| \geq c|U'|$

$\partial(U') = \{v \in N(U') \; : \; |N(v) \cap U'| = 1\}$

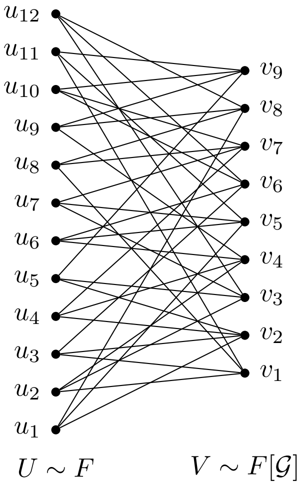
Example

- left-degree $d = 3$
- expanding set size $r = 3$
- boundary expansion factor $c = 1$

Lemma ([Raz16a])

For $\varepsilon > 0$ and n, d with $|U| = n$, $|V| = n^{\mathcal{O}(1/d)}$, $d \leq |V|^{\frac{1}{2}-\varepsilon}$, there are $(d, r, 2)$-boundary expanders $\mathcal{G}$ with $r = d \log n$

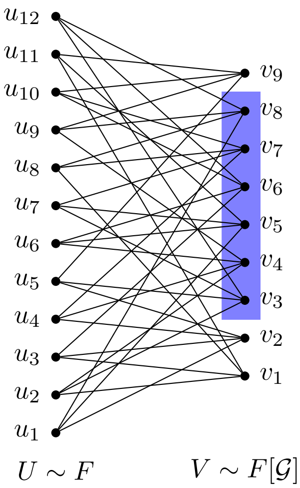
Sketch of Proof Sketch



To play on $F[\mathcal{G}]$, Player 2 simulates game on F

$\forall$ position β on $F[\mathcal{G}]$, maintain position α on F

Sketch of Proof Sketch



To play on $F[\mathcal{G}]$, Player 2 simulates game on F

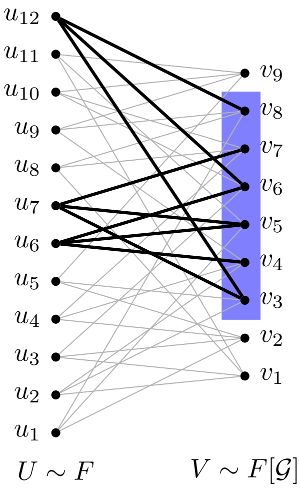
$\forall$ position β on $F[\mathcal{G}]$, maintain position α on F

Key concept: Ker(V') $= \{u \in U : N(u) \subseteq V'\}$

Example

$V' = \{v_3, \ldots, v_8\}$, Ker$(V') = \{u_6, u_7, u_{12}\}$

Sketch of Proof Sketch



To play on $F[\mathcal{G}]$, Player 2 simulates game on F

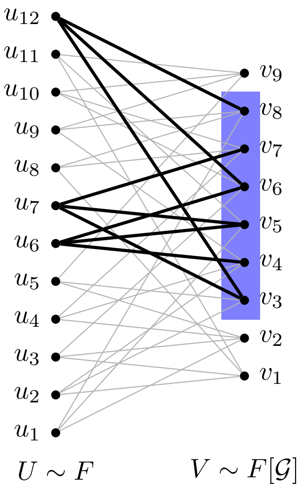
$\forall$ position β on $F[\mathcal{G}]$, maintain position α on F

Key concept: Ker$(V') = \{u \in U : N(u) \subseteq V'\}$

Example

$V' = \{v_3, \ldots, v_8\}$, Ker$(V') = \{u_6, u_7, u_{12}\}$

Sketch of Proof Sketch



To play on $F[\mathcal{G}]$, Player 2 simulates game on F
$\forall$ position β on $F[\mathcal{G}]$, maintain position α on F

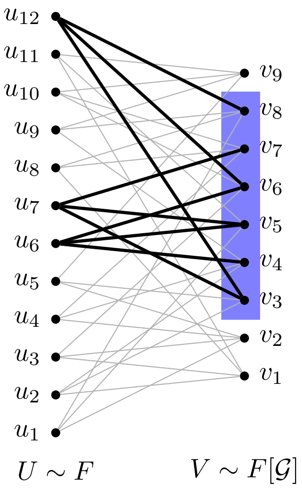
Key concept: $\mathrm{Ker}(V') = \{u \in U : N(u) \subseteq V'\}$

Example

$V' = \{v_3, \ldots, v_8\}$, $\mathrm{Ker}(V') = \{u_6, u_7, u_{12}\}$

Make sure u determined by β gets right value
$\alpha(u) = \bigoplus_{v \in N(u)} v$ — by unique neighbours

Sketch of Proof Sketch



To play on $F[\mathcal{G}]$, Player 2 simulates game on F
$\forall$ position β on $F[\mathcal{G}]$, maintain position α on F

Key concept: $\text{Ker}(V') = \{u \in U : N(u) \subseteq V'\}$

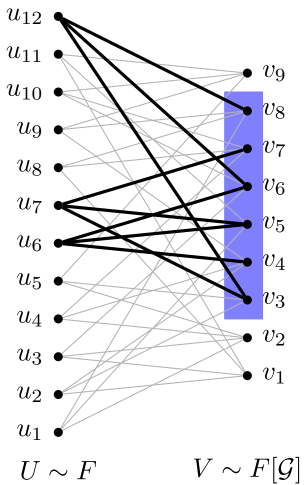
Example

$V' = \{v_3, \ldots, v_8\}$, $\text{Ker}(V') = \{u_6, u_7, u_{12}\}$

Make sure u determined by β gets right value $\alpha(u) = \bigoplus_{v \in N(u)} v$ — by unique neighbours

$|V'| \leq r \implies |\text{Ker}(V')| \leq |V'|$ by expansion, so not too many pebbles in simulated game

Sketch of Proof Sketch



To play on $F[\mathcal{G}]$, Player 2 simulates game on F
$\forall$ position β on $F[\mathcal{G}]$, maintain position α on F

Key concept: $\mathsf{Ker}(V') = \{u \in U : N(u) \subseteq V'\}$

Example

$V' = \{v_3, \ldots, v_8\}$, $\mathsf{Ker}(V') = \{u_6, u_7, u_{12}\}$

Make sure u determined by β gets right value $\alpha(u) = \bigoplus_{v \in N(u)} v$ — by unique neighbours

$|V'| \leq r \implies |\mathsf{Ker}(V')| \leq |V'|$ by expansion, so not too many pebbles in simulated game

Locally looks almost like XORification without recycling, so previous approach might work...
And give bound in terms of $|U| \gg |V|$

Hardness Condensation

Actual details more involved, but work out as follows:

Main Technical Lemma

If

- Player 2 survives R of k-game on F
- $\mathcal{G}$ is $(d, 2k, 2)$-boundary expander

then

- Player 2 survives $\frac{R}{2k}$ rounds of k-game on $F[\mathcal{G}]$

Hardness Condensation

Actual details more involved, but work out as follows:

Main Technical Lemma

If

- Player 2 survives R of k-game on F
- $\mathcal{G}$ is $(d, 2k, 2)$-boundary expander

then

- Player 2 survives $\frac{R}{2k}$ rounds of k-game on $F[\mathcal{G}]$

More about hardness condensation

- Method introduced in [Raz16a] to show that treelike resolution in bounded width k can require doubly exponential length $2^{n^{\Omega(k)}}$
- Also applied to linear programming hierarchies [Raz16c]
- Space/width trade-offs in resolution [BN16b]
- Variable space/length trade-offs [Raz16b]

Concluding Remarks

Summary

- $n^{\Omega(k/\log k)}$ lower bound on the quantifier depth of $\mathcal{L}^k$ and $\mathcal{C}^k$

Concluding Remarks

Summary

- $n^{\Omega(k/\log k)}$ lower bound on the quantifier depth of $\mathcal{L}^k$ and $\mathcal{C}^k$
- nearly matches the trivial n^{k-1} upper bound

Concluding Remarks

Summary

- $n^{\Omega(k/\log k)}$ lower bound on the quantifier depth of $\mathcal{L}^k$ and $\mathcal{C}^k$
- nearly matches the trivial n^{k-1} upper bound
- also implies near-optimal lower bound on the number of refinement steps for k-Weisfeiler–Leman

Concluding Remarks

Summary

- $n^{\Omega(k/\log k)}$ lower bound on the quantifier depth of $\mathcal{L}^k$ and $\mathcal{C}^k$
- nearly matches the trivial n^{k-1} upper bound
- also implies near-optimal lower bound on the number of refinement steps for k-Weisfeiler–Leman

Open questions

- Our result are for k-ary relational structures—prove similar lower bounds for graphs?
- Better lower bounds for XOR formulas?
- Where else can hardness condensation be useful?

Concluding Remarks

Summary

- $n^{\Omega(k/\log k)}$ lower bound on the quantifier depth of $\mathcal{L}^k$ and $\mathcal{C}^k$
- nearly matches the trivial n^{k-1} upper bound
- also implies near-optimal lower bound on the number of refinement steps for k-Weisfeiler–Leman

Open questions

- Our result are for k-ary relational structures—prove similar lower bounds for graphs?
- Better lower bounds for XOR formulas?
- Where else can hardness condensation be useful?

Thank you for your attention!

References I

[AD08] Albert Atserias and Víctor Dalmau. A combinatorial characterization of resolution width. *Journal of Computer and System Sciences*, 74(3):323–334, May 2008. Preliminary version in *CCC '03*.

[Bab16] László Babai. Graph isomorphism in quasipolynomial time. In *Proceedings of the 48th Annual ACM Symposium on Theory of Computing (STOC '16)*, pages 684–697, June 2016.

[Bar77] Jon Barwise. On Moschovakis closure ordinals. *Journal of Symbolic Logic*, 42(2):292–296, June 1977.

[BN16a] Christoph Berkholz and Jakob Nordström. Near-optimal lower bounds on quantifier depth and Weisfeiler-Leman refinement steps. In *Proceedings of the 31st Annual ACM/IEEE Symposium on Logic in Computer Science (LICS '16)*, pages 267–276, July 2016.

[BN16b] Christoph Berkholz and Jakob Nordström. Supercritical space-width trade-offs for resolution. In *Proceedings of the 43rd International Colloquium on Automata, Languages and Programming (ICALP '16)*, volume 55 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 57:1–57:14, July 2016.

References II

[CFI92] Jin-yi Cai, Martin Fürer, and Neil Immerman. An optimal lower bound on the number of variables for graph identifications. *Combinatorica*, 12(4):389–410, 1992. Preliminary version in *FOCS '89*.

[Für01] Martin Fürer. Weisfeiler–Lehman refinement requires at least a linear number of iterations. In *Proceedings of the 28th International Colloquium on Automata, Languages, and Programming (ICALP '01)*, volume 2076 of *Lecture Notes in Computer Science*, pages 322–333. Springer, July 2001.

[Gro98] Martin Grohe. Finite variable logics in descriptive complexity theory. *Bulletin of Symbolic Logic*, 4(4):345–398, 1998.

[Gro99] Martin Grohe. Equivalence in finite-variable logics is complete for polynomial time. *Combinatorica*, 19(4):507–532, October 1999.

[Gro12] Martin Grohe. Fixed-point definability and polynomial time on graphs with excluded minors. *Journal of the ACM*, 59(5):27:1–27:64, October 2012. Preliminary version in *LICS '10*.

[Hel96] Lauri Hella. Logical hierarchies in PTIME. *Information and Computation*, 129:1–19, August 1996.

References III

[IL90] Neil Immerman and Eric Lander. Describing graphs: a first-order approach to graph canonization. In Alan L. Selman, editor, *Complexity Theory Retrospective: In Honor of Juris Hartmanis on the Occasion of His Sixtieth Birthday*, pages 59–81. Springer, 1990.

[Imm81] Neil Immerman. Number of quantifiers is better than number of tape cells. *Journal of Computer and System Sciences*, 22(3):384–406, June 1981.

[Imm82] Neil Immerman. Upper and lower bounds for first order expressibility. *Journal of Computer and System Sciences*, 25(1):76–98, August 1982.

[KS16] Sandra Kiefer and Pascal Schweitzer. Upper bounds on the quantifier depth for graph differentiation in first order logic. In *Proceedings of the 31st Annual ACM/IEEE Symposium on Logic in Computer Science (LICS '16)*, pages 287–296, July 2016.

[KV15] Andreas Krebs and Oleg Verbitsky. Universal covers, color refinement, and two-variable counting logic: Lower bounds for the depth. In *Proceedings of the 30th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS '15)*, pages 689–700, July 2015.

[Raz16a] Alexander A. Razborov. A new kind of tradeoffs in propositional proof complexity. *Journal of the ACM*, 63(2):16:1–16:14, April 2016.

References IV

[Raz16b] Alexander A. Razborov. On space and depth in resolution. Technical Report TR16-184, Electronic Colloquium on Computational Complexity (ECCC), November 2016.

[Raz16c] Alexander A. Razborov. On the width of semi-algebraic proofs and algorithms. Technical Report TR16-010, Electronic Colloquium on Computational Complexity (ECCC), January 2016.

[Var95] Moshe Y. Vardi. On the complexity of bounded-variable queries (Extended abstract). In *Proceedings of the 14th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS '95)*, pages 266–276, May 1995.