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

Christoph Berkholz

Humboldt-Universität zu Berlin
&
Simons Institute for the Theory of Computing

{Symmetry, Logic, Computation}
November 2016

Joint work with Jakob Nordström (KTH Stockholm).

k-variable fragments of first-order logic

Two vertices are connected by a path of length 4:

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

k-variable fragments of first-order logic

Two vertices are connected by a path of length 4:

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

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

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

k-variable fragments of first-order logic

Two vertices are connected by a path of length 4:

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

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

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

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

k-variable fragments of first-order logic

Two vertices are connected by a path of length 4:

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

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

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

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

Vertex has degree ≥ 7:

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

k-variable fragments of first-order logic

Two vertices are connected by a path of length 4:

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

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

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

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

Vertex has degree ≥ 7:

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

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

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

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

Definition

$\mathsf{D}^k(\mathcal{A},\mathcal{B})$ is the minimal quantifier depth of a $\mathcal{C}^k$ sentence that distinguishes 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

$\mathsf{D}^k(\mathcal{A},\mathcal{B})$ is the minimal quantifier depth of a $\mathcal{C}^k$ sentence that distinguishes two n-element structures $\mathcal{A}$ and $\mathcal{B}$ (with $\mathcal{A} \not\equiv_{\mathcal{C}^k} \mathcal{B}$).

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

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

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

Definition

$\mathrm{D}^k(\mathcal{A},\mathcal{B})$ is the minimal quantifier depth of a $\mathcal{C}^k$ sentence that distinguishes 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

$\mathsf{D}^k(\mathcal{A},\mathcal{B})$ is the minimal quantifier depth of a $\mathcal{C}^k$ sentence that distinguishes two n-element structures $\mathcal{A}$ and $\mathcal{B}$ (with $\mathcal{A} \not\equiv_{\mathcal{C}^k} \mathcal{B}$).

- $\mathsf{D}^n(\mathcal{A},\mathcal{B}) \le n$

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

- $\mathsf{D}^k(\mathcal{A},\mathcal{B}) \le n^{k-1}$ $\quad \mathsf{D}^3(\mathcal{A},\mathcal{B}) \le \mathrm{O}\big(n^2/\log n\big)$ [Kiefer, Schweitzer 2016]

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

Definition

$\mathrm{D}^k(\mathcal{A}, \mathcal{B})$ is the minimal quantifier depth of a $\mathcal{C}^k$ sentence that distinguishes two n-element structures $\mathcal{A}$ and $\mathcal{B}$ (with $\mathcal{A} \not\equiv_{\mathcal{C}^k} \mathcal{B}$).

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

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

- $\mathrm{D}^k(\mathcal{A}, \mathcal{B}) \leq n^{k-1}$ $\quad \mathrm{D}^3(\mathcal{A}, \mathcal{B}) \leq \mathrm{O}\left(n^2 / \log n\right)$ [Kiefer, Schweitzer 2016]
- k const.: $\mathrm{D}^k(\mathcal{A}, \mathcal{B}) \geq \Omega\left(n\right)$ [Grohe 1996] [Fürer 2001] [Krebs, Verbitsky 2015]

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

Definition

$\mathrm{D}^k(\mathcal{A}, \mathcal{B})$ is the minimal quantifier depth of a $\mathcal{C}^k$ sentence that distinguishes two n-element structures $\mathcal{A}$ and $\mathcal{B}$ (with $\mathcal{A} \not\equiv_{\mathcal{C}^k} \mathcal{B}$).

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

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

- $\mathrm{D}^k(\mathcal{A}, \mathcal{B}) \leq n^{k-1}$ $\quad \mathrm{D}^3(\mathcal{A}, \mathcal{B}) \leq \mathrm{O}\left(n^2/\log n\right)$ [Kiefer, Schweitzer 2016]
- k const.: $\mathrm{D}^k(\mathcal{A}, \mathcal{B}) \geq \Omega\left(n\right)$ [Grohe 1996] [Fürer 2001] [Krebs, Verbitsky 2015]

Theorem [B., Nordström 2016a]

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 [B., Nordström 2016a]

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

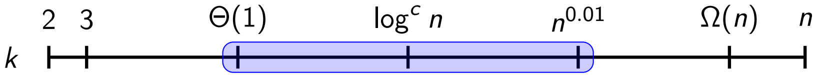
Theorem [B., Nordström 2016a]

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 [Cai, Fürer, Immerman 1992]

$\mathrm{D}^k(\mathcal{A}, \mathcal{B})$ agrees with the number of refinement steps $(k-1)$-dimensional Weisfeiler-Leman needs to distinguish $\mathcal{A}$ and $\mathcal{B}$.

$\mathcal{C}^k$ and Weisfeiler-Leman



Theorem [B., Nordström 2016a]

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 [Cai, Fürer, Immerman 1992]

$\mathrm{D}^k(\mathcal{A},\mathcal{B})$ agrees with the number of refinement steps $(k-1)$-dimensional Weisfeiler-Leman needs to distinguish $\mathcal{A}$ and $\mathcal{B}$.

An application for non-constant k

- Babai's quasipolynomial graph isomorphism test uses $k = \log^c n$ on $(k-1)$-ary relational structures. [Babai 2016]
- Our result implies an $\Omega\big(n^{\log^{c-1} n}\big)$ lower bound in this setting.

Essence of the proof

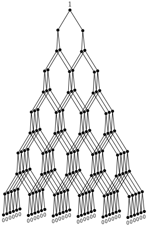
In one sentence, a novel combination of methods from

Descriptive Complexity and Proof Complexity.

Essence of the proof

In one sentence, a novel combination of methods from

Descriptive Complexity and Proof Complexity.

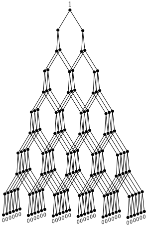


pyramid construction
[Immerman 1981]

Essence of the proof

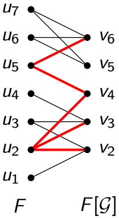
In one sentence, a novel combination of methods from

Descriptive Complexity and Proof Complexity.



pyramid construction
[Immerman 1981]

+

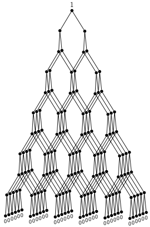


hardness condensation
[Razborov 2016a]

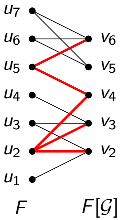
Essence of the proof

In one sentence, a novel combination of methods from

Descriptive Complexity and Proof Complexity.



$\bigoplus$



pyramid construction
[Immerman 1981]

hardness condensation
[Razborov 2016a]

The link between both areas is made via XOR formulas as a source of hard instances.

XOR formulas

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

XOR formulas

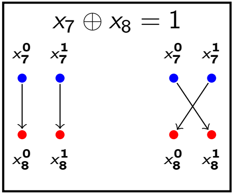
An *s-XOR formula* F over Boolean variables $x_1, \ldots, x_n$ is a set of parity constraints $x_{i_1} \oplus \cdots \oplus x_{i_r} = a$, $r \leq s$, $a \in \{0, 1\}$.

Let $\mathcal{A}(F)$ and $\mathcal{B}(F)$ relational structures with two vertices x_i^0, x_i^1 for every $x_i \in \mathrm{Vars}(F)$ and relations

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

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 F is played by two players.

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

A pebble game on XOR formulas

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

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

In each round starting from α_i:

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

A pebble game on XOR formulas

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

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

In each round starting from α_i:

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

Player 1 wins the game, if α_i falsifies an XOR-constraint.

Equivalent characterizations of the pebble game

Let F be an s-XOR formula and $R > 0, k > s$ be integers. Then the following statements are equivalent:

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

(b) There is a k-variable sentence $\varphi \in \mathcal{C}^k$ of quantifier rank R such that $\mathcal{A}(F) \models \varphi$ and $\mathcal{B}(F) \not\models \varphi$.

Equivalent characterizations of the pebble game

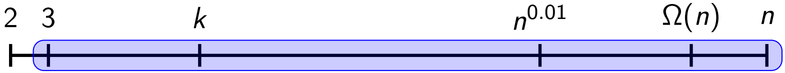
Let F be an s-XOR formula and $R > 0, k > s$ be integers. Then the following statements are equivalent:

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

(b) There is a k-variable sentence $\varphi \in \mathcal{C}^k$ of quantifier rank R such that $\mathcal{A}(F) \models \varphi$ and $\mathcal{B}(F) \not\models \varphi$.

(c) The s-CNF-formula $\mathrm{cnf}(F)$ has a resolution refutation of width $k - 1$ and depth R.

Outline of the proof



[Immerman 1981]:

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

Outline of the proof

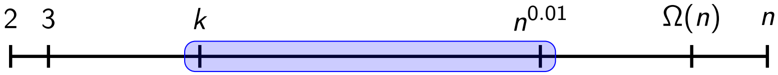
[Immerman 1981]:

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

Part I (pyramid construction):

For every k there are n-variable 3-XOR formulas such that Player 1 needs $n^{\Omega(\frac{1}{\log k})}$ rounds to win the ℓ-pebble game for $3 \leq \ell \leq k$.

Outline of the proof



[Immerman 1981]:

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

Part I (pyramid construction):

For every k there are n-variable 3-XOR formulas such that Player 1 needs $n^{\Omega(\frac{1}{\log k})}$ rounds to win the ℓ-pebble game for $3 \leq \ell \leq k$.

Part II (hardness condensation):

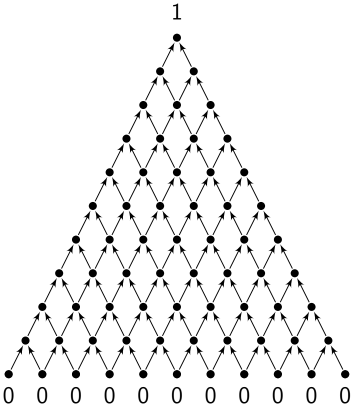
Reduce the number of variables without destroying the lower bound.

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

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

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

A two-dimensional pyramid



XORs from DAGs

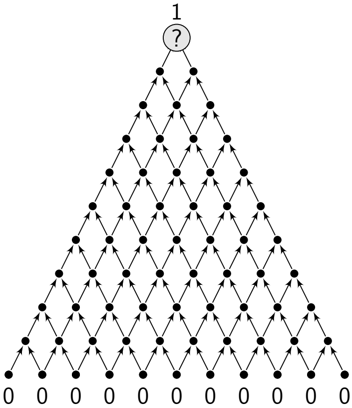
Let $\mathcal{G}$ be an acyclic directed graph with a unique sink z. The XOR-formula $\mathrm{xor}(\mathcal{G})$ over the variables $v \in V(\mathcal{G})$ contains the following constraints:

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

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

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

A two-dimensional pyramid



XORs from DAGs

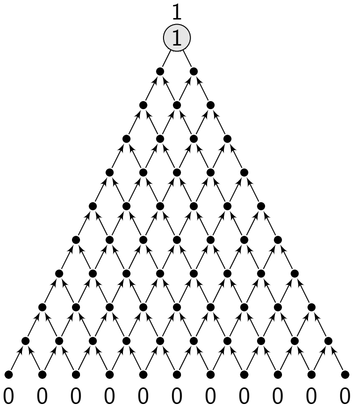
Let $\mathcal{G}$ be an acyclic directed graph with a unique sink z. The XOR-formula $\mathrm{xor}(\mathcal{G})$ over the variables $v \in V(\mathcal{G})$ contains the following constraints:

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

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

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

A two-dimensional pyramid



XORs from DAGs

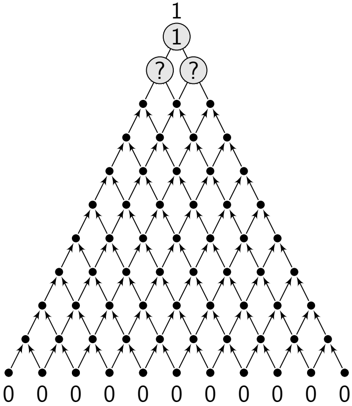
Let $\mathcal{G}$ be an acyclic directed graph with a unique sink z. The XOR-formula $\mathrm{xor}(\mathcal{G})$ over the variables $v \in V(\mathcal{G})$ contains the following constraints:

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

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

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

A two-dimensional pyramid



XORs from DAGs

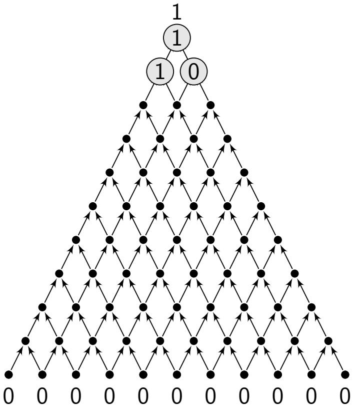
Let $\mathcal{G}$ be an acyclic directed graph with a unique sink z. The XOR-formula $\mathrm{xor}(\mathcal{G})$ over the variables $v \in V(\mathcal{G})$ contains the following constraints:

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

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

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

A two-dimensional pyramid



XORs from DAGs

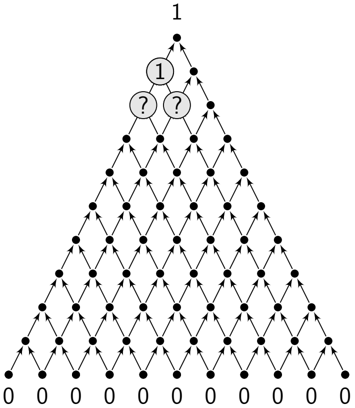
Let $\mathcal{G}$ be an acyclic directed graph with a unique sink z. The XOR-formula $\mathrm{xor}(\mathcal{G})$ over the variables $v \in V(\mathcal{G})$ contains the following constraints:

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

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

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

A two-dimensional pyramid



XORs from DAGs

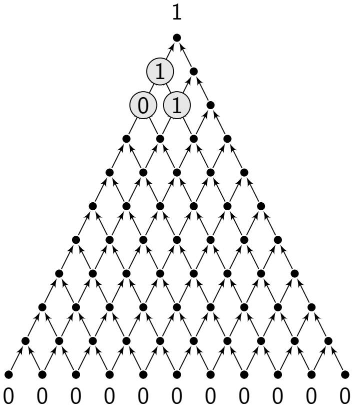
Let $\mathcal{G}$ be an acyclic directed graph with a unique sink z. The XOR-formula xor($\mathcal{G}$) over the variables $v \in V(\mathcal{G})$ contains the following constraints:

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

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

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

A two-dimensional pyramid



XORs from DAGs

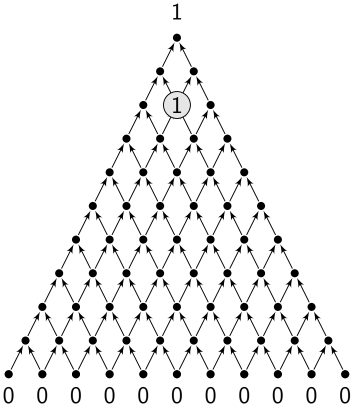
Let $\mathcal{G}$ be an acyclic directed graph with a unique sink z. The XOR-formula $\mathrm{xor}(\mathcal{G})$ over the variables $v \in V(\mathcal{G})$ contains the following constraints:

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

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

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

A two-dimensional pyramid



XORs from DAGs

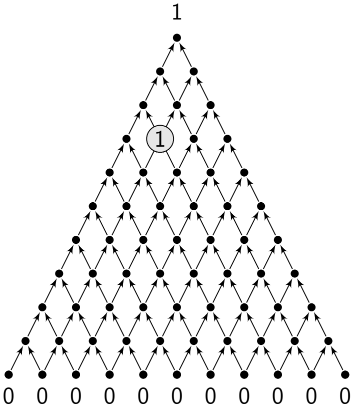
Let $\mathcal{G}$ be an acyclic directed graph with a unique sink z. The XOR-formula $\mathrm{xor}(\mathcal{G})$ over the variables $v \in V(\mathcal{G})$ contains the following constraints:

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

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

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

A two-dimensional pyramid



XORs from DAGs

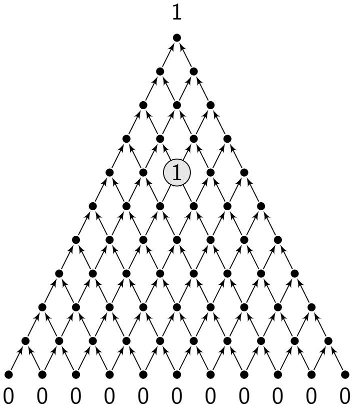
Let $\mathcal{G}$ be an acyclic directed graph with a unique sink z. The XOR-formula $\mathrm{xor}(\mathcal{G})$ over the variables $v \in V(\mathcal{G})$ contains the following constraints:

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

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

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

A two-dimensional pyramid



XORs from DAGs

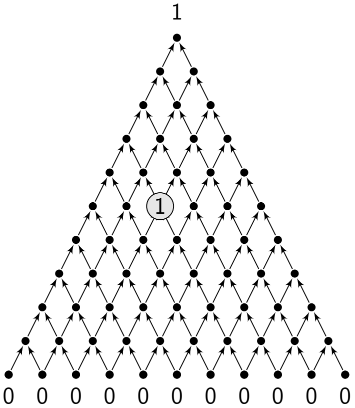
Let $\mathcal{G}$ be an acyclic directed graph with a unique sink z. The XOR-formula $\mathrm{xor}(\mathcal{G})$ over the variables $v \in V(\mathcal{G})$ contains the following constraints:

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

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

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

A two-dimensional pyramid



XORs from DAGs

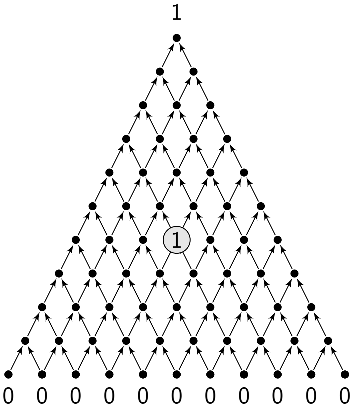
Let $\mathcal{G}$ be an acyclic directed graph with a unique sink z. The XOR-formula $\text{xor}(\mathcal{G})$ over the variables $v \in V(\mathcal{G})$ contains the following constraints:

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

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

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

A two-dimensional pyramid



XORs from DAGs

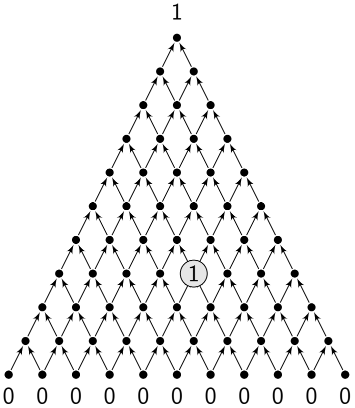
Let $\mathcal{G}$ be an acyclic directed graph with a unique sink z. The XOR-formula $\mathrm{xor}(\mathcal{G})$ over the variables $v \in V(\mathcal{G})$ contains the following constraints:

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

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

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

A two-dimensional pyramid



XORs from DAGs

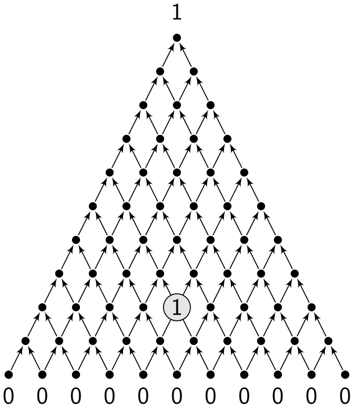
Let $\mathcal{G}$ be an acyclic directed graph with a unique sink z. The XOR-formula $\mathrm{xor}(\mathcal{G})$ over the variables $v \in V(\mathcal{G})$ contains the following constraints:

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

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

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

A two-dimensional pyramid



XORs from DAGs

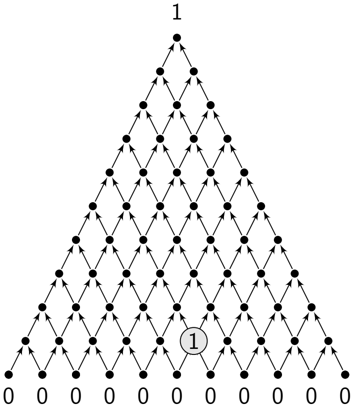
Let $\mathcal{G}$ be an acyclic directed graph with a unique sink z. The XOR-formula $\mathrm{xor}(\mathcal{G})$ over the variables $v \in V(\mathcal{G})$ contains the following constraints:

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

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

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

A two-dimensional pyramid



XORs from DAGs

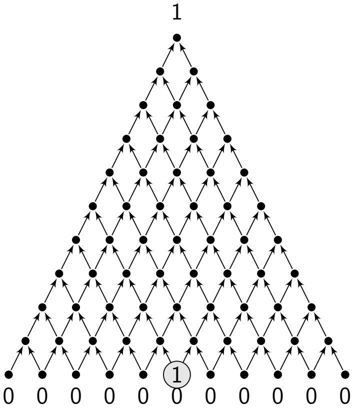
Let $\mathcal{G}$ be an acyclic directed graph with a unique sink z. The XOR-formula $\mathrm{xor}(\mathcal{G})$ over the variables $v \in V(\mathcal{G})$ contains the following constraints:

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

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

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

A two-dimensional pyramid



XORs from DAGs

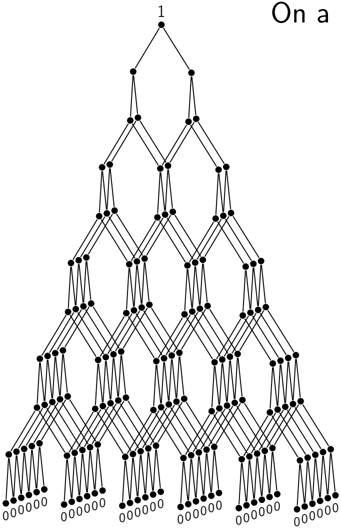
Let $\mathcal{G}$ be an acyclic directed graph with a unique sink z. The XOR-formula $\mathrm{xor}(\mathcal{G})$ over the variables $v \in V(\mathcal{G})$ contains the following constraints:

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

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

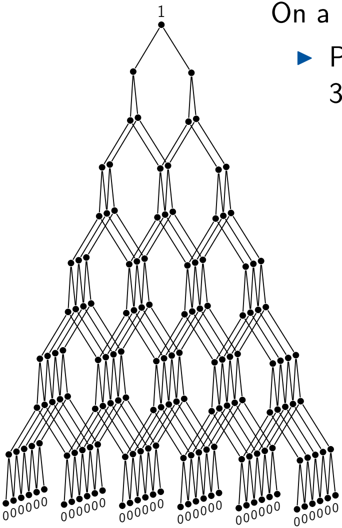
(iii) $z = 1$ for the unique sink z.

A three-dimensional pyramid



On a d-dimensional pyramid of height h

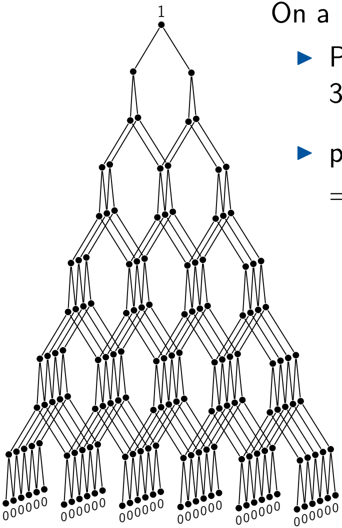
A three-dimensional pyramid



On a 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 three-dimensional pyramid



On a 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

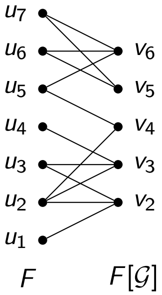
PART II: Hardness condensation

Substitution with recycling

Let F be an XOR-formula over variables U and $\mathcal{G} = (U \,\dot\cup\, V, E)$ a bipartite graph.
The formula $F[\mathcal{G}]$ is defined over variables V with constraints

$$\Big(\bigoplus_{v \in N(u_1)} v\Big) \oplus \cdots \oplus \Big(\bigoplus_{v \in N(u_\ell)} v\Big) = a$$

for all constraints $u_1 \oplus \cdots \oplus u_\ell = a$ from F.

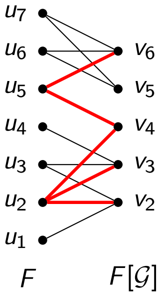


Substitution with recycling

Let F be an XOR-formula over variables U and $\mathcal{G} = (U \mathbin{\dot\cup} V, E)$ a bipartite graph.
The formula $F[\mathcal{G}]$ is defined over variables V with constraints

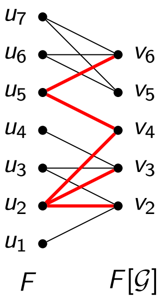
$$\left(\bigoplus_{v \in N(u_1)} v\right) \oplus \cdots \oplus \left(\bigoplus_{v \in N(u_\ell)} v\right) = a$$

for all constraints $u_1 \oplus \cdots \oplus u_\ell = a$ from F.



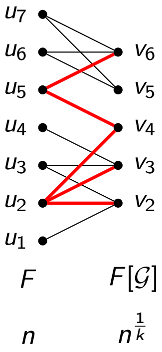
$$u_2 \oplus u_5 = 0 \longrightarrow v_2 \oplus v_3 \oplus v_4 \oplus v_4 \oplus v_6 = 0$$

Substitution with recycling



$$u_2 \oplus u_5 = 0 \longrightarrow v_2 \oplus v_3 \oplus v_4 \oplus v_4 \oplus v_6 = 0$$

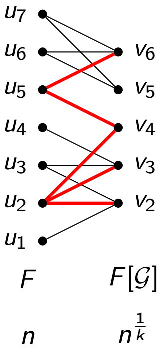
Substitution with recycling



$$u_2 \oplus u_5 = 0 \longrightarrow v_2 \oplus v_3 \oplus v_4 \oplus v_4 \oplus v_6 = 0$$

- We use boundary expander $\mathcal{G}$ with left-degree $< k/3$, $|U| = n$, and $|V| \approx n^{\frac{1}{k}}$.

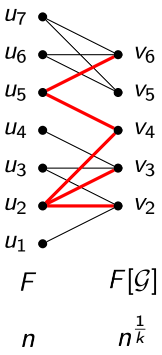
Substitution with recycling



$$u_2 \oplus u_5 = 0 \longrightarrow v_2 \oplus v_3 \oplus v_4 \oplus v_4 \oplus v_6 = 0$$

- We use boundary expander $\mathcal{G}$ with left-degree $< k/3$, $|U| = n$, and $|V| \approx n^{\frac{1}{k}}$.
- Player 1 wins the k-pebble game on $F[\mathcal{G}]$.

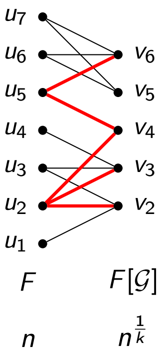
Substitution with recycling



$$u_2 \oplus u_5 = 0 \longrightarrow v_2 \oplus v_3 \oplus v_4 \oplus v_4 \oplus v_6 = 0$$

- We use boundary expander $\mathcal{G}$ with left-degree $< k/3$, $|U| = n$, and $|V| \approx n^{\frac{1}{k}}$.
- Player 1 wins the k-pebble game on $F[\mathcal{G}]$.
- If Player 2 survives the R-round k-pebble game on F, then she survives the $(\approx R)$-round k-pebble game on $F[\mathcal{G}]$.

Substitution with recycling



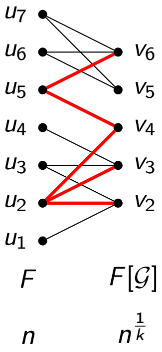
$$u_2 \oplus u_5 = 0 \longrightarrow v_2 \oplus v_3 \oplus v_4 \oplus v_4 \oplus v_6 = 0$$

- We use boundary expander $\mathcal{G}$ with left-degree $< k/3$, $|U| = n$, and $|V| \approx n^{\frac{1}{k}}$.
- Player 1 wins the k-pebble game on $F[\mathcal{G}]$.
- If Player 2 survives the R-round k-pebble game on F, then she survives the $(\approx R)$-round k-pebble game on $F[\mathcal{G}]$.

Remark

This method was introduced in [Razborov 2016a] to show that treelike resolution refutations of bounded width k require doubly exponential length $2^{n^{\Omega(k)}}$.

Substitution with recycling



$$u_2 \oplus u_5 = 0 \longrightarrow v_2 \oplus v_3 \oplus v_4 \oplus v_4 \oplus v_6 = 0$$

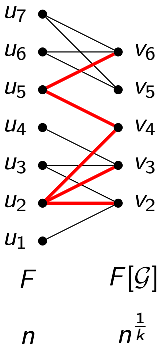
- We use boundary expander $\mathcal{G}$ with left-degree $< k/3$, $|U| = n$, and $|V| \approx n^{\frac{1}{k}}$.
- Player 1 wins the k-pebble game on $F[\mathcal{G}]$.
- If Player 2 survives the R-round k-pebble game on F, then she survives the $(\approx R)$-round k-pebble game on $F[\mathcal{G}]$.

Remark

This method was introduced in [Razborov 2016a] to show that treelike resolution refutations of bounded width k require doubly exponential length $2^{n^{\Omega(k)}}$.

Later applied to linear programming hierarchies [Razborov 2016b]

Substitution with recycling



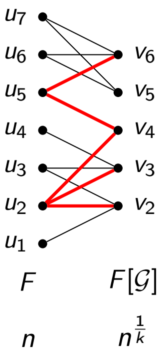
$$u_2 \oplus u_5 = 0 \longrightarrow v_2 \oplus v_3 \oplus v_4 \oplus v_4 \oplus v_6 = 0$$

- We use boundary expander $\mathcal{G}$ with left-degree $< k/3$, $|U| = n$, and $|V| \approx n^{\frac{1}{k}}$.
- Player 1 wins the k-pebble game on $F[\mathcal{G}]$.
- If Player 2 survives the R-round k-pebble game on F, then she survives the $(\approx R)$-round k-pebble game on $F[\mathcal{G}]$.

Remark

This method was introduced in [Razborov 2016a] to show that treelike resolution refutations of bounded width k require doubly exponential length $2^{n^{\Omega(k)}}$.
Later applied to linear programming hierarchies [Razborov 2016b], space/width trade-offs in resolution [B., Nordström 2016b]

Substitution with recycling



$$u_2 \oplus u_5 = 0 \longrightarrow v_2 \oplus v_3 \oplus v_4 \oplus v_4 \oplus v_6 = 0$$

- We use boundary expander $\mathcal{G}$ with left-degree $< k/3$, $|U| = n$, and $|V| \approx n^{\frac{1}{k}}$.
- Player 1 wins the k-pebble game on $F[\mathcal{G}]$.
- If Player 2 survives the R-round k-pebble game on F, then she survives the ($\approx R$)-round k-pebble game on $F[\mathcal{G}]$.

Remark

This method was introduced in [Razborov 2016a] to show that treelike resolution refutations of bounded width k require doubly exponential length $2^{n^{\Omega(k)}}$.
Later applied to linear programming hierarchies [Razborov 2016b], space/width trade-offs in resolution [B., Nordström 2016b], variable space/length trade-offs [Razborov 2016c].

Conclusion

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

Conclusion

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

Conclusion

- $n^{\Omega(\frac{k}{\log k})}$ 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-WL

Conclusion

- $n^{\Omega(\frac{k}{\log k})}$ 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-WL

Open Question

Our examples are k-ary relational structures. Are there similar lower bounds for graphs?

Conclusion

- $n^{\Omega(\frac{k}{\log k})}$ 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-WL

Open Question

Our examples are k-ary relational structures. Are there similar lower bounds for graphs?

Thank you for your attention!