Graph Colouring Is Hard on Average for Polynomial Calculus and Nullstellensatz

Jakob Nordström

University of Copenhagen and Lund University

Milestones and Motifs in the Theory of Proofs, Algebraic Computation, and Lower Bounds
IIT Gandhinagar
December 14–15, 2024

Joint work with Jonas Conneryd, Susanna de Rezende, Shuo Pang, and Kilian Risse

Graph Colouring Is Hard on Average for Polynomial Calculus and Nullstellensatz

Jakob Nordström

University of Copenhagen and Lund University

Milestones and Motifs in the Theory of Proofs, Algebraic Computation, and Lower Bounds
IIT Gandhinagar
December 14–15, 2024

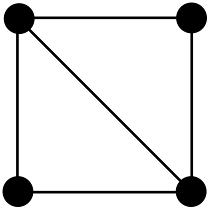
Joint work with Jonas Conneryd, Susanna de Rezende, Shuo Pang, and Kilian Risse

Thanks for the slides!

Graph Colouring

Can vertices of graph G be coloured with k colours so that all neighbours get distinct colours?

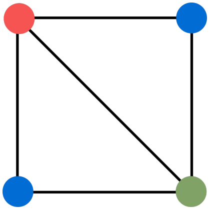
One of Karp's 21 NP-complete problems



Graph Colouring

Can vertices of graph G be coloured with k colours so that all neighbours get distinct colours?

One of Karp's 21 NP-complete problems



Is Graph Colouring Hard?

Colouring seems hard even to approximate:

- If G k-colourable, best efficient algorithm uses $k \cdot \widetilde{\Omega}(n)$ colours [Halldorsson 93]
- If G 3-colourable, best algorithm uses $n^{0.199\cdots}$ colours [Kawarabayashi–Thorup 17]
- NP-hard to approximate within factor $n^{1-\varepsilon}$ [Feige–Kilian 98, Zuckerman 07]

Is Graph Colouring Hard?

Colouring seems hard even to approximate:

- If G k-colourable, best efficient algorithm uses $k \cdot \widetilde{\Omega}(n)$ colours [Halldorsson 93]
- If G 3-colourable, best algorithm uses $n^{0.199\cdots}$ colours [Kawarabayashi–Thorup 17]
- NP-hard to approximate within factor $n^{1-\varepsilon}$ [Feige–Kilian 98, Zuckerman 07]

However, applied algorithms appear to do well:

- Backtracking and SAT-based algorithms
  [San Segundo 12, Hebrard–Katsirelos 20, Heule–Karahalios–van Hoeve 22]
- Integer programming
  [Mehortra–Trick 95, Gualandi–Malucelli 12]
- Algebraic algorithms
  [DeLoera–Lee–Malkin–Margulies 08 & 11, DeLoera–Lee–Margulies–Onn 09, DeLoera–Margulies–Pernpeinter–Riedl–Rolnick–Spencer–Stasi–Swenson 15]

Is Graph Colouring Hard?

Colouring seems hard even to approximate:

- If G k-colourable, best efficient algorithm uses $k \cdot \widetilde{\Omega}(n)$ colours [Halldorsson 93]
- If G 3-colourable, best algorithm uses $n^{0.199\cdots}$ colours [Kawarabayashi–Thorup 17]
- NP-hard to approximate within factor $n^{1-\varepsilon}$ [Feige–Kilian 98, Zuckerman 07]

However, applied algorithms appear to do well:

- Backtracking and SAT-based algorithms
  [San Segundo 12, Hebrard–Katsirelos 20, Heule–Karahalios–van Hoeve 22]
- Integer programming
  [Mehortra–Trick 95, Gualandi–Malucelli 12]
- Algebraic algorithms
  [DeLoera–Lee–Malkin–Margulies 08 & 11, DeLoera–Lee–Margulies–Onn 09, DeLoera–Margulies–Pernpeinter–Riedl–Rolnick–Spencer–Stasi–Swenson 15]

Can we prove that k-colouring is hard for these algorithms?

Hardness for Algebraic Algorithms

- Exponential lower bounds known for explicit graphs
  [Lauria–Nordström 17, Atserias–Ochremiak 19]
- But obtained by reduction from other problems
- Graph colouring instances somewhat artificial

Hardness for Algebraic Algorithms

- Exponential lower bounds known for explicit graphs

  [Lauria–Nordström 17, Atserias–Ochremiak 19]
- But obtained by reduction from other problems
- Graph colouring instances somewhat artificial

Perhaps graph colouring is *easy on most graphs?*

Hardness for Algebraic Algorithms

- Exponential lower bounds known for explicit graphs
  [Lauria–Nordström 17, Atserias–Ochremiak 19]
- But obtained by reduction from other problems
- Graph colouring instances somewhat artificial

Perhaps graph colouring is *easy on most graphs?*

To rule this out, want average-case hardness results

SAT-based algorithms **[Beame–Culberson–Mitchell–Moore 05]**

Conflict-driven clause learning (CDCL) SAT solvers need exponential time for k-colouring on random graphs for $k \geq 3$

Our Result

Theorem

Algorithms based on Hilbert Nullstellensatz and/or Gröbner bases require exponential time to solve k-colouring on random graphs for $k \geq 3$

Our Result

Theorem

Algorithms based on Hilbert Nullstellensatz and/or Gröbner bases require exponential time to solve k-colouring on random graphs for $k \geq 3$

Established via proof complexity:

- Formalise reasoning method in algorithm as a proof system
- Fast execution for non-k-colourable graph G yields short proof of statement "G is not k-colourable"
- Show that such short proofs do not exist

Nullstellensatz Proof System

To show polynomials $p_1, \ldots, p_m$ in $\mathbb{F}[\vec{x}]$, have no common root in $\mathbb{F}$, suffices to find polynomials $q_1, \ldots, q_m$ in $\mathbb{F}[\vec{x}]$ such that

$$\sum_{i=1}^{m} q_i(\vec{x}) \cdot p_i(\vec{x}) = 1$$

This is a Nullstellensatz proof of unsatisfiability

[Beame–Impagliazzo–Krajíček–Pitassi–Pudlák 96]

Nullstellensatz Proof System

To show polynomials $p_1, \ldots, p_m$ in $\mathbb{F}[\vec{x}]$, have no common root in $\mathbb{F}$, suffices to find polynomials $q_1, \ldots, q_m$ in $\mathbb{F}[\vec{x}]$ such that

$$\sum_{i=1}^{m} q_i(\vec{x}) \cdot p_i(\vec{x}) = 1$$

This is a Nullstellensatz proof of unsatisfiability

[Beame–Impagliazzo–Krajíček–Pitassi–Pudlák 96]

Soundness: if such polynomials q_i exist, then clearly $\{p_i\}$ have no common root

Completeness (Boolean variables): special case of Hilbert's Nullstellensatz

Polynomial Calculus Proof System [Clegg–Edmonds–Impagliazzo 96]

Dynamic version: given $\{p_1, \ldots, p_m\}$, derive new polynomials using two rules

(linear combination)	$$\frac{p \quad q}{\alpha p + \beta q}$$	$\alpha, \beta \in \mathbb{F}$
(multiplication)	$$\frac{p}{x \cdot p}$$	x variable

Goal: derive polynomial 1

Polynomial Calculus Proof System [Clegg–Edmonds–Impagliazzo 96]

Dynamic version: given $\{p_1, \ldots, p_m\}$, derive new polynomials using two rules

$$\text{(linear combination)} \quad \frac{p \quad q}{\alpha p + \beta q} \qquad \alpha, \beta \in \mathbb{F}$$

$$\text{(multiplication)} \quad \frac{p}{x \cdot p} \qquad x \text{ variable}$$

Goal: derive polynomial 1

Polynomial calculus proof system models Gröbner basis computations

Polynomial Calculus Proof System [Clegg–Edmonds–Impagliazzo 96]

Dynamic version: given $\{p_1, \ldots, p_m\}$, derive new polynomials using two rules

$$\text{(linear combination)} \quad \frac{p \quad q}{\alpha p + \beta q} \qquad \alpha, \beta \in \mathbb{F}$$

$$\text{(multiplication)} \quad \frac{p}{x \cdot p} \qquad x \text{ variable}$$

Goal: derive polynomial 1

Polynomial calculus proof system models Gröbner basis computations

- **Proof size:** # of monomials in derivation
  Make proof system stronger by allowing dual variables $\overline{x}_i$ for negative literals
  [Alekhnovich–Ben-Sasson–Razborov–Wigderson 02]
- **Proof degree:** max total degree of polynomial in derivation

Encoding k-Colouring as Polynomials

Variables $x_{v,i}$ = "vertex v gets colour i", $v \in V(G)$, $i \in [k]$

Axiom polynomials for graph G:

Each vertex gets a colour	$\sum_{i=1}^{k} x_{v,i} - 1$	
Colours are unique	$x_{v,i} \cdot x_{v,i'}$	$i \neq i'$
Distinct colours for neighbours	$x_{u,i} \cdot x_{v,i}$	$(u,v) \in E(G)$
Variables are Boolean	$x_{v,i}^2 - x_{v,i}$	

Encoding k-Colouring as Polynomials

Variables $x_{v,i}$ = "vertex v gets colour i", $v \in V(G)$, $i \in [k]$

Axiom polynomials for graph G:

Each vertex gets a colour	$\sum_{i=1}^{k} x_{v,i} - 1$	
Colours are unique	$x_{v,i} \cdot x_{v,i'}$	$i \neq i'$
Distinct colours for neighbours	$x_{u,i} \cdot x_{v,i}$	$(u,v) \in E(G)$
Variables are Boolean	$x_{v,i}^2 - x_{v,i}$	

Common root of polynomials $\Leftrightarrow$ k-colouring of G

Encoding k-Colouring as Polynomials

Variables $x_{v,i}$ = "vertex v gets colour i", $v \in V(G)$, $i \in [k]$

Axiom polynomials for graph G:

Each vertex gets a colour	$\sum_{i=1}^{k} x_{v,i} - 1$	
Colours are unique	$x_{v,i} \cdot x_{v,i'}$	$i \neq i'$
Distinct colours for neighbours	$x_{u,i} \cdot x_{v,i}$	$(u, v) \in E(G)$
Variables are Boolean	$x_{v,i}^2 - x_{v,i}$	

Common root of polynomials $\Leftrightarrow$ k-colouring of G

Other important encoding used in computational algebra [Bayer 82]:

- Colours X_v are kth roots of unity $\{1, \zeta, \zeta^2, \cdots, \zeta^{k-1}\}$ (assuming $\mathrm{char}(\mathbb{F}) \nmid k$)
- Linear substitution from X_v to $x_{v,1}, \ldots, x_{v,k}$ $\Rightarrow$ (roughly) same proof degree

More Formal Statement of Result

Theorem

For G random sparse graph on n vertices, with probability $1 - o(1)$ any polynomial calculus proof of fact "G is not 3-colourable" has size $\exp(\Omega(n))$

More Formal Statement of Result

Theorem

For G random sparse graph on n vertices, with probability $1 - o(1)$ any polynomial calculus proof of fact "G is not 3-colourable" has size $\exp(\Omega(n))$

- Lower bound holds over any field
- For both random regular graphs and Erdős–Rényi random graphs (with appropriately chosen parameters)

More Formal Statement of Result

Theorem

For G random sparse graph on n vertices, with probability $1 - o(1)$ any polynomial calculus proof of fact "G is not 3-colourable" has size $\exp(\Omega(n))$

- Lower bound holds over any field
- For both random regular graphs and Erdős–Rényi random graphs (with appropriately chosen parameters)
- Obtained by showing $\Omega(n)$ degree lower bound
- Implies exponential size lower bound for Boolean encoding

[Impagliazzo–Pudlák–Sgall 99]

Degree Lower Bound: Framework

Task: separate 1 from $\{$polynomials derivable in degree $D\}$

Degree Lower Bound: Framework

Task: separate 1 from {polynomials derivable in degree D}

[Razborov 98]: suffices to find linear $R : \mathbb{F}[\vec{x}] \longrightarrow \mathbb{F}[\vec{x}]$ such that

1. $R(\text{axiom}) = 0$
2. $R(xp) = R(xR(p))$ for any p of degree $\leq D - 1$
3. $R(1) \neq 0$

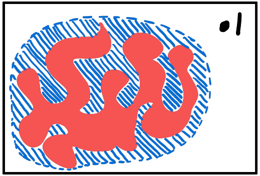
R overapproximates what is derivable in degree D

Degree Lower Bound: Framework

Task: separate 1 from {polynomials derivable in degree D}

[Razborov 98]: suffices to find linear $R : \mathbb{F}[\vec{x}] \to \mathbb{F}[\vec{x}]$ such that

1. $R(\text{axiom}) = 0$
2. $R(xp) = R(xR(p))$ for any p of degree $\leq D - 1$
3. $R(1) \neq 0$



Derivable in degree D

$\ker(R)$

R overapproximates what is derivable in degree D

Quick Recap: Polynomial Ideals

Given set of polynomials $\mathcal{P}$, ideal $\langle \mathcal{P} \rangle$ is smallest set such that

- $\mathcal{P} \subseteq \langle \mathcal{P} \rangle$
- $p, q \in \langle \mathcal{P} \rangle \Rightarrow p + q \in \langle \mathcal{P} \rangle$
- $p \in \langle \mathcal{P} \rangle \Rightarrow r \cdot p \in \langle \mathcal{P} \rangle$ for all polynomials r

Quick Recap: Polynomial Ideals

Given set of polynomials $\mathcal{P}$, ideal $\langle \mathcal{P} \rangle$ is smallest set such that

- $\mathcal{P} \subseteq \langle \mathcal{P} \rangle$
- $p, q \in \langle \mathcal{P} \rangle \Rightarrow p + q \in \langle \mathcal{P} \rangle$
- $p \in \langle \mathcal{P} \rangle \Rightarrow r \cdot p \in \langle \mathcal{P} \rangle$ for all polynomials r

Connection to polynomial calculus:

- $\langle \mathcal{P} \rangle$ contains all polynomial implied by $\mathcal{P}$
- Which is exactly what is derivable by polynomial calculus
- $1 \in \langle \mathcal{P} \rangle \Leftrightarrow \mathcal{P}$ is unsatisfiable

Polynomial Ideal Reductions

- Impose total order on monomials (with 1 smallest)
- Order polynomials by largest monomial (leading monomial)
- **Reduction modulo ideal $\langle \mathcal{P} \rangle$:** Operator $R_{\langle \mathcal{P} \rangle} : \mathbb{F}[\vec{x}] \to \mathbb{F}[\vec{x}]$ defined as

$$R_{\langle \mathcal{P} \rangle}(q) := \text{minimum polynomial in } \{q - r \mid r \in \langle \mathcal{P} \rangle\}$$

Polynomial Ideal Reductions

- Impose total order on monomials (with 1 smallest)
- Order polynomials by largest monomial (leading monomial)
- **Reduction modulo ideal $\langle \mathcal{P} \rangle$:** Operator $R_{\langle \mathcal{P} \rangle} : \mathbb{F}[\vec{x}] \to \mathbb{F}[\vec{x}]$ defined as

$$R_{\langle \mathcal{P} \rangle}(q) := \text{minimum polynomial in } \{q - r \mid r \in \langle \mathcal{P} \rangle\}$$

Properties of $R_{\langle \mathcal{P} \rangle}$:

- well-defined
- linear
- $\ker\left(R_{\langle \mathcal{P} \rangle}\right) = \langle \mathcal{P} \rangle$
- $R^2_{\langle \mathcal{P} \rangle} = R_{\langle \mathcal{P} \rangle}$

Example of Polynomial Reduction

Consider $\mathbb{F}[x, y]$ and ideal generated by $\{x + y\}$.

- Order $x \succ y$ extended to all monomials (lexicographically, say)
- $\mathcal{R}_{\langle x+y \rangle} : \; x^a y^b \mapsto (-1)^a y^{a+b}$

Pseudo-reductions

Reduction operator $R_{\langle \mathcal{P} \rangle}$ satisfies properties postulated by Razborov!

Pseudo-reductions

Reduction operator $R_{\langle \mathcal{P} \rangle}$ satisfies properties postulated by Razborov!
Except $R_{\langle \mathcal{P} \rangle}(1) = 0$, since $\mathcal{P}$ unsatisfiable…
So won't get degree lower bounds from reduction modulo $\langle \mathcal{P} \rangle$

Pseudo-reductions

Reduction operator $R_{\langle \mathcal{P} \rangle}$ satisfies properties postulated by Razborov!
Except $R_{\langle \mathcal{P} \rangle}(1) = 0$, since $\mathcal{P}$ unsatisfiable...
So won't get degree lower bounds from reduction modulo $\langle \mathcal{P} \rangle$

Fix: reduce modulo smaller ideals!

Alekhnovich-Razborov 03

- For each monomial m, reduce m modulo ideal of subset $S(m)$ of axioms
- Extend to polynomials by linearity

Pseudo-reductions

Reduction operator $R_{\langle \mathcal{P} \rangle}$ satisfies properties postulated by Razborov!
Except $R_{\langle \mathcal{P} \rangle}(1) = 0$, since $\mathcal{P}$ unsatisfiable…
So won't get degree lower bounds from reduction modulo $\langle \mathcal{P} \rangle$

Fix: reduce modulo smaller ideals!

Alekhnovich-Razborov 03

- For each monomial m, reduce m modulo ideal of subset $S(m)$ of axioms
- Extend to polynomials by linearity

Intuition:

- $S(m)$ contains axioms "closely related" to variables in m
- R indistinguishable from polynomial ideal reduction in low degree, but $R(1) \neq 0$
- Think of R as pseudo-reduction modulo fake ideal claiming that $\mathcal{P}$ is satisfiable

From Pseudo-reductions to Degree Lower Bounds

Recall that we want three properties from linear operator R:

1. $R(\text{axiom}) = 0$
2. $R(xp) = R(xR(p))$ for any p of degree $\leq D - 1$
3. $R(1) \neq 0$

From Pseudo-reductions to Degree Lower Bounds

Recall that we want three properties from linear operator R:

1. $R(\text{axiom}) = 0$
2. $R(xp) = R(xR(p))$ for any p of degree $\leq D - 1$
3. $R(1) \neq 0$

This would show:

- All input axioms in $\mathcal{P}$ are in $\ker(R)$
- All polynomials derivable from $\mathcal{P}$ in degree $\leq D$ are in $\ker(R)$
- But $1 \notin \ker(R)$
- So degree lower bound $> D$ follows

Getting Pseudo-reductions to Behave Well

- Concretely, for axiom polynomial $p = m_1 + m_2$ want $R(p) = 0$

Getting Pseudo-reductions to Behave Well

- Concretely, for axiom polynomial $p = m_1 + m_2$ want $R(p) = 0$
- But pseudo-reduction

$$R(p) = R(m_1) + R(m_2) = R_{\langle S(m_1)\rangle}(m_1) + R_{\langle S(m_2)\rangle}(m_2)$$

reduces monomials modulo different ideals — lose control of what happens

Getting Pseudo-reductions to Behave Well

- Concretely, for axiom polynomial $p = m_1 + m_2$ want $R(p) = 0$
- But pseudo-reduction

$$R(p) = R(m_1) + R(m_2) = R_{\langle S(m_1)\rangle}(m_1) + R_{\langle S(m_2)\rangle}(m_2)$$

  reduces monomials modulo different ideals — lose control of what happens
- Dream scenario: Show that there exists ideal $\mathcal{I}$ such that
  - $p \in \mathcal{I}$
  - $S(m_i) \subseteq \mathcal{I}$ for $i = 1, 2$
  - $R_{\langle S(m_i)\rangle}(m_i) = R_{\mathcal{I}}(m_i)$ for $i = 1, 2$

Getting Pseudo-reductions to Behave Well

- Concretely, for axiom polynomial $p = m_1 + m_2$ want $R(p) = 0$
- But pseudo-reduction

$$R(p) = R(m_1) + R(m_2) = R_{\langle S(m_1)\rangle}(m_1) + R_{\langle S(m_2)\rangle}(m_2)$$

  reduces monomials modulo different ideals — lose control of what happens
- Dream scenario: Show that there exists ideal $\mathcal{I}$ such that
  - $p \in \mathcal{I}$
  - $S(m_i) \subseteq \mathcal{I}$ for $i = 1, 2$
  - $R_{\langle S(m_i)\rangle}(m_i) = R_{\mathcal{I}}(m_i)$ for $i = 1, 2$
- Then

$$R(p) = R_{\langle S(m_1)\rangle}(m_1) + R_{\langle S(m_2)\rangle}(m_2) = R_{\mathcal{I}}(m_1) + R_{\mathcal{I}}(m_2) = R_{\mathcal{I}}(m_1 + m_2) = 0$$

Why Aren't We Done Already?

- All of this is old news...
  - Proposed in [Alekhnovich–Razborov 03]
  - Further developed in, e.g., [Galesi–Lauria 10a, 10b; Mikša–Nordström 15]

Why Aren't We Done Already?

- All of this is old news…
  - Proposed in [Alekhnovich–Razborov 03]
  - Further developed in, e.g., [Galesi–Lauria 10a, 10b; Mikša–Nordström 15]
- Technical crux: Requires finding subset of axioms that can be "nicely isolated"
  - For pigeonhole principle (PHP), if some pigeons assigned to holes, residual problem is still PHP instance
  - But partial colouring propagates constraints throughout whole graph!?

Why Aren't We Done Already?

- All of this is old news...
  - Proposed in [Alekhnovich–Razborov 03]
  - Further developed in, e.g., [Galesi–Lauria 10a, 10b; Mikša–Nordström 15]
- Technical crux: Requires finding subset of axioms that can be "nicely isolated"
  - For pigeonhole principle (PHP), if some pigeons assigned to holes, residual problem is still PHP instance
  - But partial colouring propagates constraints throughout whole graph!?
- Average-case polynomial calculus lower bounds for colouring remained open since [Beame–Culberson–Mitchell–Moore 05]

Why Aren't We Done Already?

- All of this is old news...
  - Proposed in [Alekhnovich–Razborov 03]
  - Further developed in, e.g., [Galesi–Lauria 10a, 10b; Mikša–Nordström 15]
- Technical crux: Requires finding subset of axioms that can be "nicely isolated"
  - For pigeonhole principle (PHP), if some pigeons assigned to holes, residual problem is still PHP instance
  - But partial colouring propagates constraints throughout whole graph!?
- Average-case polynomial calculus lower bounds for colouring remained open since [Beame–Culberson–Mitchell–Moore 05]
- Crucial new ideas in [Romero-Tunçel 22] — more about that later

Degree Lower Bounds for Colouring

- For colouring, associate to each monomial m a vertex set V_m
- Define

$$R\left(\sum_i c_i m_i \right) := \sum_i c_i \underbrace{R_{V_{m_i}}}_{\text{reduction modulo ideal of “}G[V_{m_i}]\text{ is }k\text{-colourable”}} (m_i)$$

- Technical challenge: construct V_m so that R satisfies required properties

Vertex Set V_m

Say that monomial $m = x_{u,2}x_{v,3}x_{w,1}$ mentions vertices u, v, w

Vertex Set V_m

Say that monomial $m = x_{u,2}x_{v,3}x_{w,1}$ mentions vertices u, v, w

Vertices V_m related to m

- Define closure $\mathrm{Cl}(U) \supseteq U$ of vertex sets U
- Set $V_m := \mathrm{Cl}(\{\text{vertices mentioned in } m\})$

Vertex Set V_m

Say that monomial $m = x_{u,2}x_{v,3}x_{w,1}$ mentions vertices u, v, w

Vertices V_m related to m

- Define closure $\mathrm{Cl}(U) \supseteq U$ of vertex sets U
- Set $V_m := \mathrm{Cl}(\{\text{vertices mentioned in } m\})$

Desired properties of closure:

1. **Subset-preserving:** $U' \subseteq \mathrm{Cl}(U) \Rightarrow \mathrm{Cl}(U') \subseteq \mathrm{Cl}(U)$
2. **Size-preserving:** $|U| \leq D \Rightarrow |\mathrm{Cl}(U)| = O(D)$
3. **Reduction-preserving:** For any monomial m mentioning only vertices in $\mathrm{Cl}(U)$ and any vertex set J of size $O(D)$ it holds that

$$R_{\mathrm{Cl}(U)}(m) = R_{\mathrm{Cl}(U) \cup J}(m)$$

Vertex Set V_m

Say that monomial $m = x_{u,2}x_{v,3}x_{w,1}$ mentions vertices u, v, w

Vertices V_m related to m

- Define closure $\mathrm{Cl}(U) \supseteq U$ of vertex sets U
- Set $V_m := \mathrm{Cl}(\{\text{vertices mentioned in } m\})$

Desired properties of closure:

1. **Subset-preserving:** $U' \subseteq \mathrm{Cl}(U) \Rightarrow \mathrm{Cl}(U') \subseteq \mathrm{Cl}(U)$
2. **Size-preserving:** $|U| \leq D \Rightarrow |\mathrm{Cl}(U)| = O(D)$
3. **Reduction-preserving:** For any monomial m mentioning only vertices in $\mathrm{Cl}(U)$ and any vertex set J of size $O(D)$ it holds that

$$R_{\mathrm{Cl}(U)}(m) = R_{\mathrm{Cl}(U) \cup J}(m)$$

Reduction-Preserving Property of Closure

Reduction-preserving: For any monomial m mentioning only vertices in $\mathrm{Cl}(U)$ and any vertex set J of size $O(D)$ it holds that $R_{\mathrm{Cl}(U)}(m) = R_{\mathrm{Cl}(U) \cup J}(m)$

Reduction-Preserving Property of Closure

Reduction-preserving: For any monomial m mentioning only vertices in $\mathrm{Cl}(U)$ and any vertex set J of size $O(D)$ it holds that $R_{\mathrm{Cl}(U)}(m) = R_{\mathrm{Cl}(U)\cup J}(m)$

Reduction lemma [CdRNPR 23]

For fixed order on vertices (and variables), can achieve this property if:

- each colouring of $G[\mathrm{Cl}(U)]$ can be extended to $G[\mathrm{Cl}(U) \cup J]$

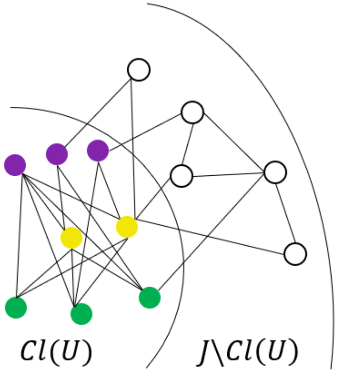
Reduction-Preserving Property of Closure

Reduction-preserving: For any monomial m mentioning only vertices in $\mathrm{Cl}(U)$ and any vertex set J of size $O(D)$ it holds that $R_{\mathrm{Cl}(U)}(m) = R_{\mathrm{Cl}(U) \cup J}(m)$

Reduction lemma [CdRNPR 23]

For fixed order on vertices (and variables), can achieve this property if:

- each colouring of $G[\mathrm{Cl}(U)]$ can be extended to $G[\mathrm{Cl}(U) \cup J]$



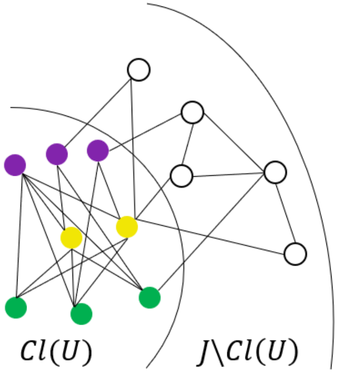
Reduction-Preserving Property of Closure

Reduction-preserving: For any monomial m mentioning only vertices in $\mathsf{Cl}(U)$ and any vertex set J of size $O(D)$ it holds that $R_{\mathsf{Cl}(U)}(m) = R_{\mathsf{Cl}(U)\cup J}(m)$

Reduction lemma [CdRNPR 23]

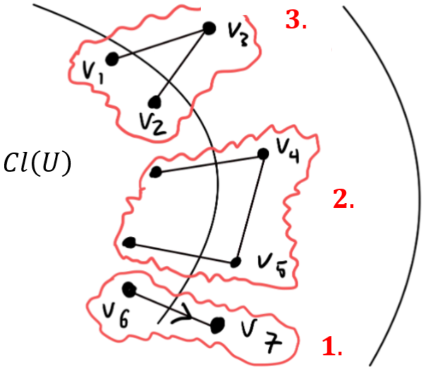
For fixed order on vertices (and variables), can achieve this property if:

- each colouring of $G[\mathsf{Cl}(U)]$ can be extended to $G[\mathsf{Cl}(U) \cup J]$
- ... in **order-decreasing** way: for each v in $J \setminus \mathsf{Cl}(U)$, colour can be determined based on colouring of $\{w \in \mathsf{Cl}(U) : w < v\}$



Construction of Closure (1/2)

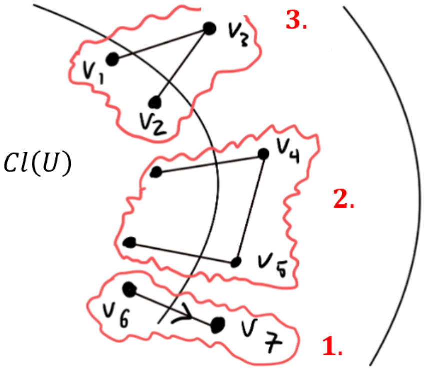
Sufficient to prevent certain structures in neighbourhood outside Cl(U)



Construction of Closure (1/2)

Sufficient to prevent certain structures in neighbourhood outside $\mathrm{Cl}(U)$

1. Vertex with a larger neighbour in $\mathrm{Cl}(U)$
2. Edge between neighbours of $\mathrm{Cl}(U)$
3. Vertex with $>$ one neighbours in $\mathrm{Cl}(U)$

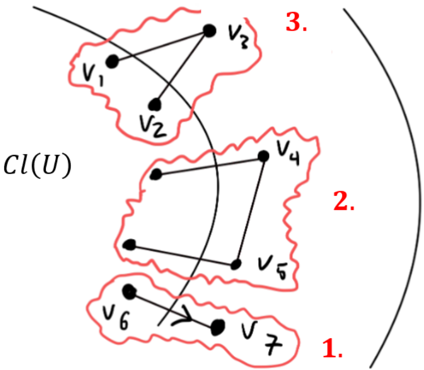


Construction of Closure (1/2)

Sufficient to prevent certain structures in neighbourhood outside $\mathsf{Cl}(U)$

1. Vertex with a larger neighbour in $\mathsf{Cl}(U)$
2. Edge between neighbours of $\mathsf{Cl}(U)$
3. Vertex with $>$ one neighbours in $\mathsf{Cl}(U)$

Same structures identified in [Romero-Tunçel 22] *in colouring lower bound for large-girth graphs!*

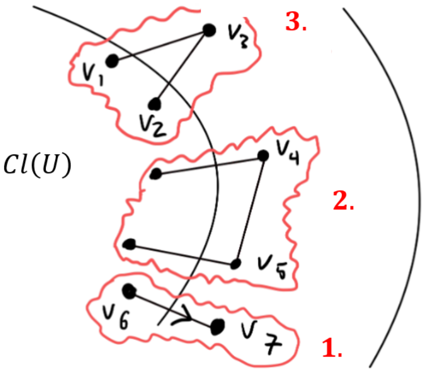


Construction of Closure (2/2)

Sufficient to prevent certain structures in neighbourhood outside Cl(U)

Constructing the closure of a set U

1 Start with given set U

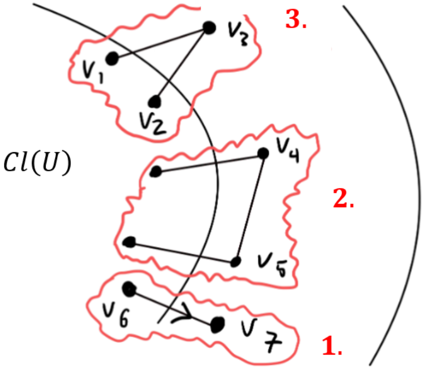


Construction of Closure (2/2)

Sufficient to prevent certain structures in neighbourhood outside $\mathrm{Cl}(U)$

Constructing the closure of a set U

1. Start with given set U
2. Add all vertices reachable from current set by order-decreasing paths in G

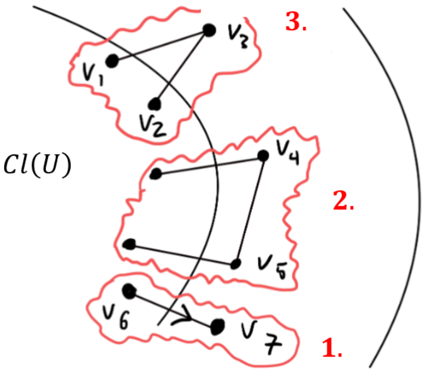


Construction of Closure (2/2)

Sufficient to prevent certain structures in neighbourhood outside Cl(U)

Constructing the closure of a set U

1. Start with given set U
2. Add all vertices reachable from current set by order-decreasing paths in G
3. If type 2 or 3 structure, add offending vertices to current set and go to **2**



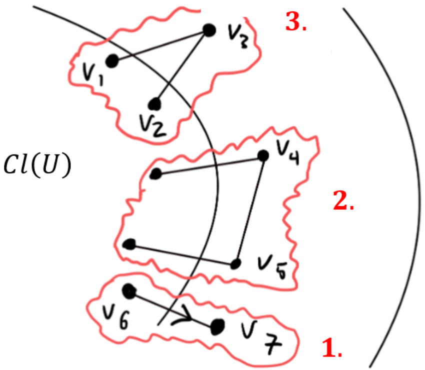
Construction of Closure (2/2)

Sufficient to prevent certain structures in neighbourhood outside $\mathrm{Cl}(U)$

Constructing the closure of a set U

1. Start with given set U
2. Add all vertices reachable from current set by order-decreasing paths in G
3. If type 2 or 3 structure, add offending vertices to current set and go to **2**

Let $\mathrm{Cl}(U)$:= final set



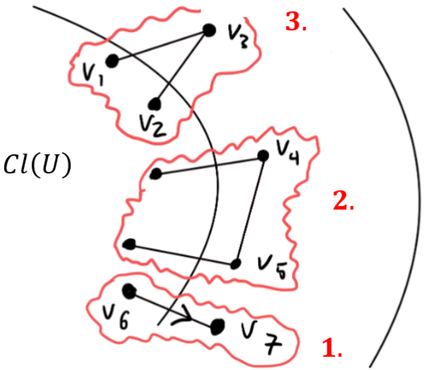
Construction of Closure (2/2)

Sufficient to prevent certain structures in neighbourhood outside $\mathrm{Cl}(U)$

Constructing the closure of a set U

1. Start with given set U
2. Add all vertices reachable from current set by order-decreasing paths in G
3. If type 2 or 3 structure, add offending vertices to current set and go to **2**

Let $\mathrm{Cl}(U) :=$ final set



Not hard to show $\mathrm{Cl}(U)$ well-defined

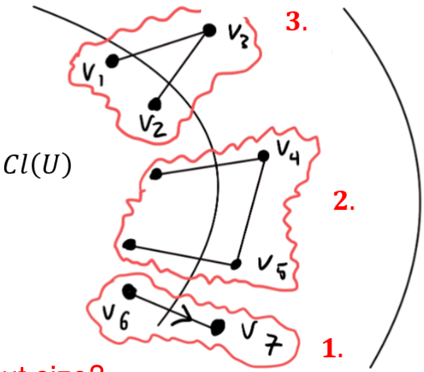
Construction of Closure (2/2)

Sufficient to prevent certain structures in neighbourhood outside Cl(U)

Constructing the closure of a set U

1. Start with given set U
2. Add all vertices reachable from current set by order-decreasing paths in G
3. If type 2 or 3 structure, add offending vertices to current set and go to **2**

Let Cl(U) := final set



Not hard to show Cl(U) well-defined, but what about size?

Keeping the Closure Small Enough

Size lemma [CdRNPR 23]

For random n-vertex graph with max vertex degree d, it holds for any vertex set U with $|U| \leq 2^{-d^{O(1)}} \cdot n$ that

$$|\mathrm{Cl}(U)| = O(|U|)$$

Keeping the Closure Small Enough

Size lemma [CdRNPR 23]

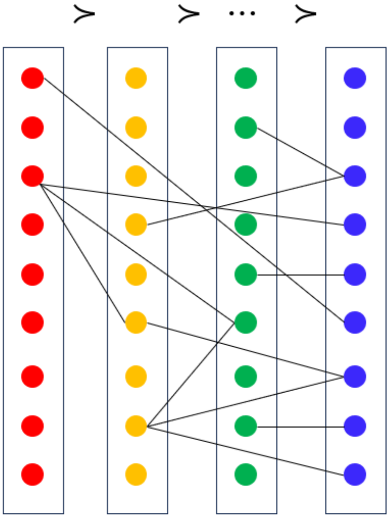
For random n-vertex graph with max vertex degree d, it holds for any vertex set U with $|U| \le 2^{-d^{O(1)}} \cdot n$ that

$$|\mathrm{Cl}(U)| = O(|U|)$$

- Proof relies on "good" vertex order introduced by [Romero-Tunçel 22]:

  Order vertices according to a valid colouring of G

- Chromatic number of random graph G is $\chi(G) = O(d/\log d) = O(1)$
  $\Longrightarrow$ order-decreasing paths have length $O(1)$



Use any vertex order that respects colour classes

Completing the Proof (Sketch) of the Colouring Lower Bound

Size lemma: $|\mathrm{Cl}(U)| = O(|U|)$ for all U of small size

- Intuition: Closure $\mathrm{Cl}(U)$ obtained from sequence of vertex sets $U \subset U_1 \subset U_2 \subset \ldots$ of increasing edge density
- But random graph has bounded edge density everywhere
  $\Rightarrow$ construction has to stop in $O(1)$ rounds, so $|\mathrm{Cl}(U)| \leq \left(d^{\chi(G)}\right)^{O(1)} \cdot |U|$

Completing the Proof (Sketch) of the Colouring Lower Bound

Size lemma: $|\mathrm{Cl}(U)| = O(|U|)$ for all U of small size

- Intuition: Closure $\mathrm{Cl}(U)$ obtained from sequence of vertex sets $U \subset U_1 \subset U_2 \subset \ldots$ of increasing edge density
- But random graph has bounded edge density everywhere
  $\Rightarrow$ construction has to stop in $O(1)$ rounds, so $|\mathrm{Cl}(U)| \leq \left(d^{\chi(G)}\right)^{O(1)} \cdot |U|$

Pseudo-reduction operator properties:

- $R(\text{axiom}) = 0$ since each axiom p mentions vertex set U_p of size ≤ 2 and $R(m) = R_{\mathrm{Cl}(U_p)}(m)$ for each monomial m in p
- $R(xp) = R(xR(p))$ for all p of degree $\leq D - 1$ since closure is size- and reduction-preserving
- $R(1) = 1$ since $\mathrm{Cl}(\emptyset) = \emptyset$ and $R_{\mathrm{Cl}(1)}(\cdot)$ hence does nothing

Some Future Research Directions

1 Colouring lower bounds for other proof systems
   - Sherali–Adams
   - Sum-of-squares
   - Cutting planes

Some Future Research Directions

1. Colouring lower bounds for other proof systems
   - Sherali–Adams
   - Sum-of-squares
   - Cutting planes
2. Polynomial calculus lower bounds for other problems
   - Graph homomorphism (generalization of colouring)
   - Clique
   - Dense linear order principle

Some Future Research Directions

1. Colouring lower bounds for other proof systems
   - Sherali–Adams
   - Sum-of-squares
   - Cutting planes
2. Polynomial calculus lower bounds for other problems
   - Graph homomorphism (generalization of colouring)
   - Clique
   - Dense linear order principle
3. Unified understanding of polynomial calculus lower bound techniques
   - Including lower bounds depending on field characteristic

Some Future Research Directions

1. Colouring lower bounds for other proof systems
   - Sherali–Adams
   - Sum-of-squares
   - Cutting planes
2. Polynomial calculus lower bounds for other problems
   - Graph homomorphism (generalization of colouring)
   - Clique
   - Dense linear order principle
3. Unified understanding of polynomial calculus lower bound techniques
   - Including lower bounds depending on field characteristic
4. Connections between pseudo-reductions and other lower bound operators
   - Designs for Nullstellensatz
   - Pseudo-expectations for sums-of-squares

Summing up

- Graph colouring is a notoriously hard problem in theory
- But applied algorithms can work surprisingly well in practice
- Can we rule out unconditionally that they solve NP-complete problems in polynomial time?

Summing up

- Graph colouring is a notoriously hard problem in theory
- But applied algorithms can work surprisingly well in practice
- Can we rule out unconditionally that they solve NP-complete problems in polynomial time?
- **Our result:** Linear degree lower bounds for polynomial calculus proofs for random graphs
- Implies exponential average-case running times for state-of-the-art algebraic algorithms

Summing up

- Graph colouring is a notoriously hard problem in theory
- But applied algorithms can work surprisingly well in practice
- Can we rule out unconditionally that they solve NP-complete problems in polynomial time?
- **Our result:** Linear degree lower bounds for polynomial calculus proofs for random graphs
- Implies exponential average-case running times for state-of-the-art algebraic algorithms
- Lots of good open problems for algebraic and semi-algebraic proof systems!

Summing up

- Graph colouring is a notoriously hard problem in theory
- But applied algorithms can work surprisingly well in practice
- Can we rule out unconditionally that they solve NP-complete problems in polynomial time?
- **Our result:** Linear degree lower bounds for polynomial calculus proofs for random graphs
- Implies exponential average-case running times for state-of-the-art algebraic algorithms
- Lots of good open problems for algebraic and semi-algebraic proof systems!

Thank you for your attention!