Gadgets and Anti-Gadgets Leading to a Complexity Dichotomy

Tyson Williams
University of Wisconsin-Madison

Joint with:
Jin-Yi Cai (University of Wisconsin-Madison)
Michael Kowalczyk (Northern Michigan University)

To appear at ITCS 2012

#VertexCover

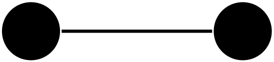
Definition

A vertex cover of a graph is a set of vertices such that each edge of the graph is incident to at least one vertex in the set.

#VertexCover

Definition

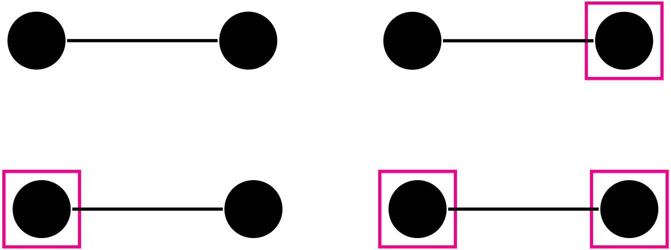
A vertex cover of a graph is a set of vertices such that each edge of the graph is incident to at least one vertex in the set.



#VertexCover

Definition

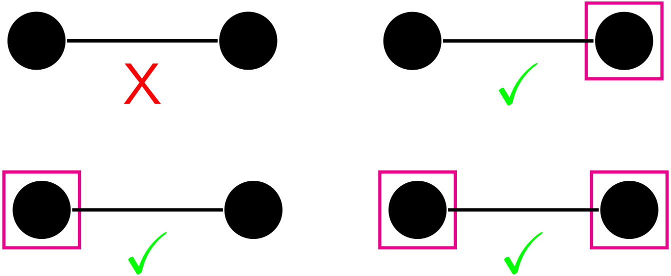
A vertex cover of a graph is a set of vertices such that each edge of the graph is incident to at least one vertex in the set.



#VertexCover

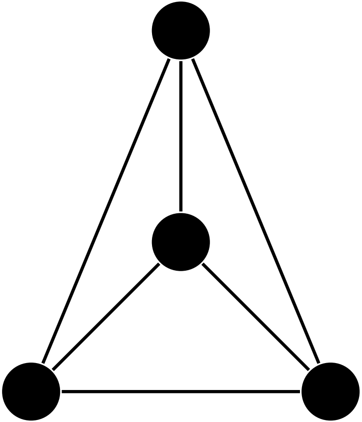
Definition

A vertex cover of a graph is a set of vertices such that each edge of the graph is incident to at least one vertex in the set.



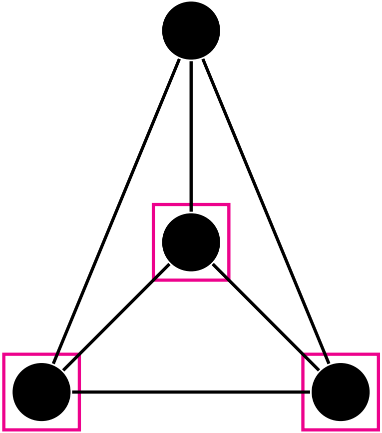
Systematic Approach to #VertexCover

- $G = (V, E)$



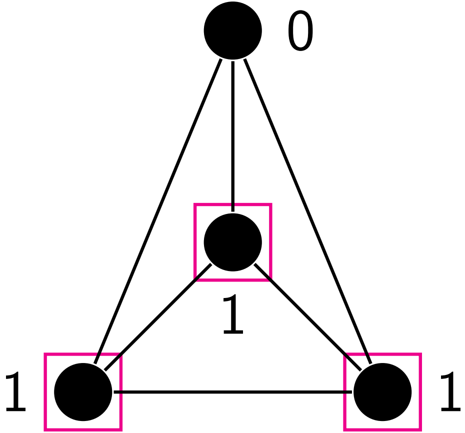
Systematic Approach to #VertexCover

- $G = (V, E)$



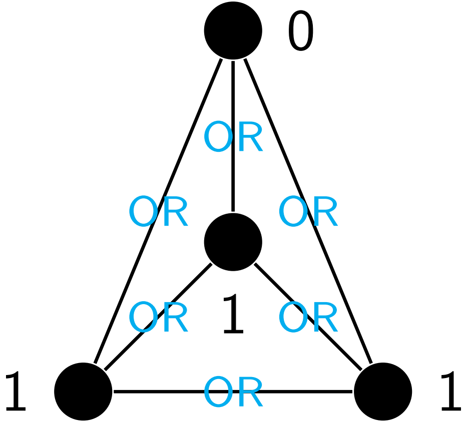
Systematic Approach to #VertexCover

- $G = (V, E)$
- $\sigma : V \to \{0, 1\}$



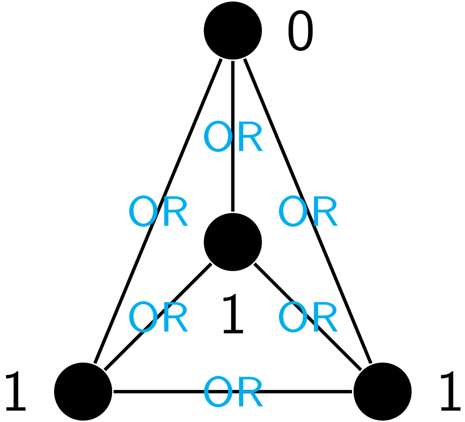
Systematic Approach to #VERTEXCOVER

- $G = (V, E)$
- $\sigma : V \to \{0, 1\}$



Systematic Approach to #VertexCover

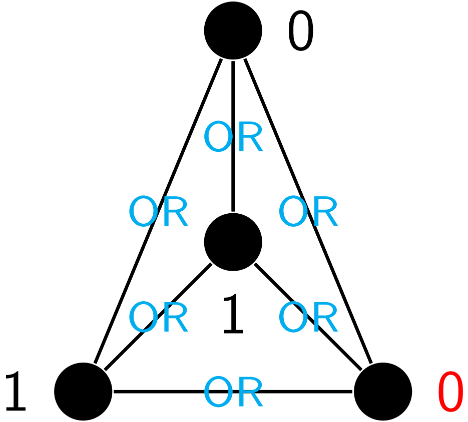
- $G = (V, E)$
- $\sigma : V \to \{0, 1\}$



$$\prod_{(u,v)\in E} \mathrm{OR}(\sigma(u), \sigma(v)) = 1 \cdot 1 \cdot 1 \cdot 1 \cdot 1 \cdot 1 = 1$$

Systematic Approach to #VertexCover

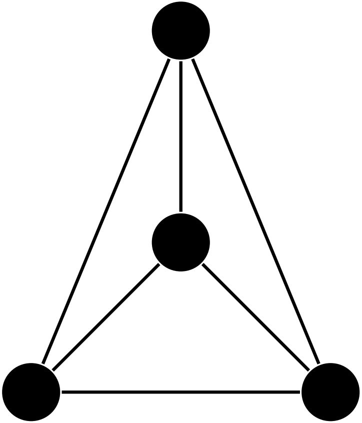
- $G = (V, E)$
- $\sigma : V \to \{0, 1\}$



$$\prod_{(u,v)\in E} \mathrm{OR}(\sigma(u), \sigma(v)) = 1 \cdot 1 \cdot 0 \cdot 1 \cdot 1 \cdot 1 = 0$$

Systematic Approach to #$\textsc{VertexCover}$

- $G = (V, E)$
- $\sigma : V \to \{0, 1\}$



$$\#\textsc{VertexCover}(G) = \sum_{\sigma : V \to \{0,1\}} \prod_{(u,v) \in E} \mathrm{OR}(\sigma(u), \sigma(v))$$

Generalize

$$\sum_{\sigma: V \to \{0,1\}} \prod_{(u,v) \in E} \mathrm{OR}\left(\sigma(u), \sigma(v)\right)$$

Generalize

$$\sum_{\sigma:V\to\{0,1\}} \prod_{(u,v)\in E} \mathrm{OR}\left(\sigma(u),\sigma(v)\right)$$

Input		Output
p	q	$\mathrm{OR}(p,q)$
0	0	0
0	1	1
1	0	1
1	1	1

Generalize

$$\sum_{\sigma:V\to\{0,1\}} \prod_{(u,v)\in E} f(\sigma(u),\sigma(v))$$

Input		Output
p	q	$\mathrm{OR}(p,q)$
0	0	0
0	1	1
1	0	1
1	1	1

Input		Output
p	q	$f(p,q)$
0	0	w
0	1	x
1	0	y
1	1	z

where $w, x, y, z \in \mathbb{C}$

Generalize

Partition Function: $Z(\cdot)$

$$Z(G) = \sum_{\sigma : V \to \{0,1\}} \prod_{(u,v) \in E} f(\sigma(u), \sigma(v))$$

Input		**Output**
p	q	$\mathrm{OR}(p,q)$
0	0	0
0	1	1
1	0	1
1	1	1

Input		**Output**
p	q	$f(p,q)$
0	0	w
0	1	x
1	0	y
1	1	z

where $w, x, y, z \in \mathbb{C}$

Main Result

Theorem (Dichotomy Theorem)

Over 3-regular graphs G, the counting problem for any (binary) complex-weighted function f

$$Z(G) = \sum_{\sigma : V \to \{0,1\}} \prod_{(u,v) \in E} f(\sigma(u), \sigma(v))$$

is either computable in polynomial time or $\#\mathrm{P}$-hard.

Main Result

Theorem (Dichotomy Theorem)

Over 3-regular graphs G, the counting problem for any (binary) complex-weighted function f

$$Z(G) = \sum_{\sigma : V \to \{0,1\}} \prod_{(u,v) \in E} f(\sigma(u), \sigma(v))$$

is either computable in polynomial time or $\#\mathrm{P}$-hard. Furthermore, the complexity is efficiently decidable.

Outline

1. Related work
2. Define Holant function
3. Proof sketch
   - Anti-gadgets

Related Work: Dichotomy Theorems

- Symmetric f
  - $f(0,1) = f(1,0)$
- 3-regular graphs with outputs in
  - $\{0,1\}$ [Cai, Lu, Xia 08]
  - $\{0,1,-1\}$ [Kowalczyk 09]
  - $\mathbb{R}$ [Cai, Lu, Xia 09]
  - $\mathbb{C}$ [Cai, Kowalczyk 10]
- k-regular graphs with outputs in
  - $\mathbb{R}$ [Cai, Kowalczyk 10]
  - $\mathbb{C}$ [Cai, Kowalczyk 11]

Related Work: Dichotomy Theorems

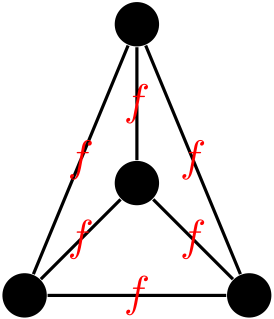
- Symmetric f
  - $f(0,1) = f(1,0)$
- 3-regular graphs with outputs in
  - $\{0,1\}$ [Cai, Lu, Xia 08]
  - $\{0,1,-1\}$ [Kowalczyk 09]
  - $\mathbb{R}$ [Cai, Lu, Xia 09]
  - $\mathbb{C}$ [Cai, Kowalczyk 10]
- k-regular graphs with outputs in
  - $\mathbb{R}$ [Cai, Kowalczyk 10]
  - $\mathbb{C}$ [Cai, Kowalczyk 11]

This work:

- Asymmetric f
- 3-regular graphs with outputs in
  - $\mathbb{C}$

Definition of Holant Function

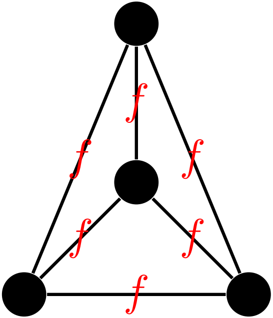
- Partition Function



$$\sum_{\sigma:V\to\{0,1\}} \prod_{(u,v)\in E} f\left(\sigma(u),\sigma(v)\right)$$

Definition of Holant Function

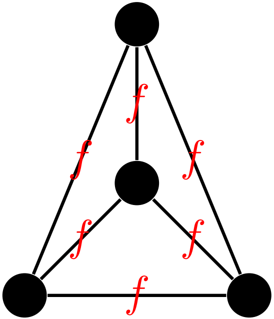
- Partition Function
  - Assignments to vertices
  - Functions on edges



$$\sum_{\sigma:V\to\{0,1\}} \prod_{(u,v)\in E} f(\sigma(u),\sigma(v))$$

Definition of Holant Function

- Partition Function
  - Assignments to vertices
  - Functions on edges

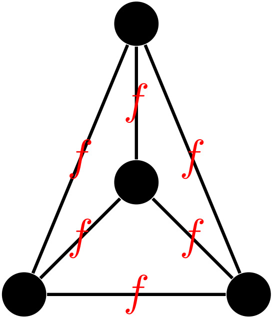


$$\sum_{\sigma:V\to\{0,1\}} \prod_{(u,v)\in E} f\left(\sigma(u),\sigma(v)\right)$$

- Holant Function
  - Assignment to edges
  - Functions on vertices

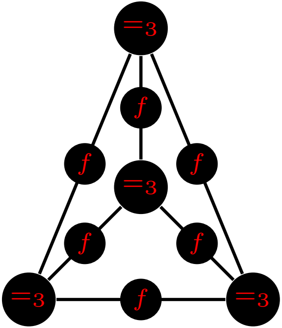
Definition of Holant Function

- Partition Function
  - Assignments to vertices
  - Functions on edges



$$\sum_{\sigma:V\to\{0,1\}} \prod_{(u,v)\in E} f\left(\sigma(u),\sigma(v)\right)$$

- Holant Function
  - Assignment to edges
  - Functions on vertices

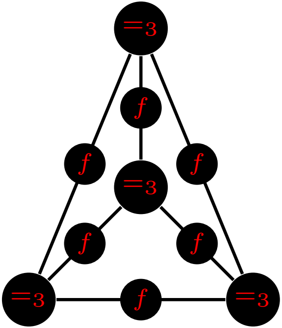


$$\sum_{\sigma:E\to\{0,1\}} \prod_{v\in V} g_v\left(\sigma\mid_{E(v)}\right)$$

Definition of Holant Function

- $\text{Holant}(\{{\color{red}f}\} \mid \{=_3\})$ is a counting problem defined over (2,3)-regular bipartite graphs.

- Holant Function
  - Assignment to edges
  - Functions on vertices

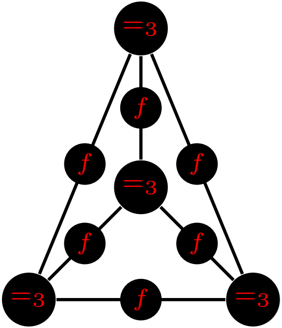


$$\sum_{\sigma: E \to \{0,1\}} \prod_{v \in V} g_v \left(\sigma \mid_{E(v)} \right)$$

Definition of Holant Function

- $\mathrm{Holant}(\{f\} \mid \{=_3\})$ is a counting problem defined over (2,3)-regular bipartite graphs.
- Degree 2 vertices take f.
- Degree 3 vertices take $=_3$.

- Holant Function
  - Assignment to edges
  - Functions on vertices



$$\sum_{\sigma:E\to\{0,1\}} \prod_{v\in V} g_v \left(\sigma \mid_{E(v)}\right)$$

Example Holant Problems

- $\text{Holant}(\{\text{OR}_2\} \mid \{=_3\})$ is #VertexCover on 3-regular graphs.

Example Holant Problems

- $\text{Holant}(\{\text{OR}_2\} \mid \{=_3\})$ is $\#\textsc{VertexCover}$ on 3-regular graphs.
- $\text{Holant}(\{\text{NAND}_2\} \mid \{=_3\})$ is $\#\textsc{IndependentSet}$ on 3-regular graphs.

Example Holant Problems

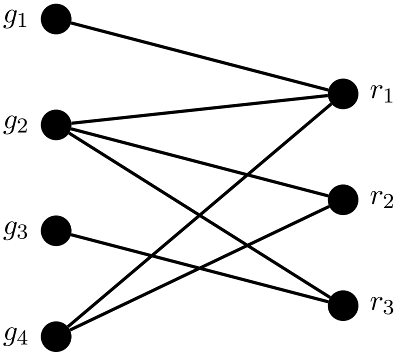
- $\mathrm{Holant}(\{\mathrm{OR}_2\} \mid \{=_3\})$ is $\#\textsc{VertexCover}$ on 3-regular graphs.
- $\mathrm{Holant}(\{\mathrm{NAND}_2\} \mid \{=_3\})$ is $\#\textsc{IndependentSet}$ on 3-regular graphs.
- $\mathrm{Holant}(\{=_2\} \mid \{\text{AT-MOST-ONE}\})$ is $\#\textsc{Matching}$.

Example Holant Problems

- $\mathrm{Holant}(\{\mathrm{OR}_2\} \,|\, \{=_3\})$ is $\#\textsc{VertexCover}$ on 3-regular graphs.
- $\mathrm{Holant}(\{\mathrm{NAND}_2\} \,|\, \{=_3\})$ is $\#\textsc{IndependentSet}$ on 3-regular graphs.
- $\mathrm{Holant}(\{=_2\} \,|\, \{\text{AT-MOST-ONE}\})$ is $\#\textsc{Matching}$.
- $\mathrm{Holant}(\{=_2\} \,|\, \{\text{EXACTLY-ONE}\})$ is $\#\textsc{PerfectMatching}$.

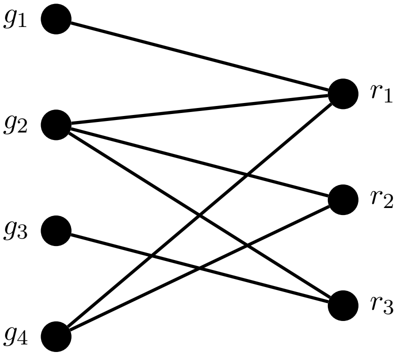
General Bipartite Holant Definition

- More generally, $\mathrm{Holant}(\mathcal{G} \,|\, \mathcal{R})$ is a counting problem defined over bipartite graphs.



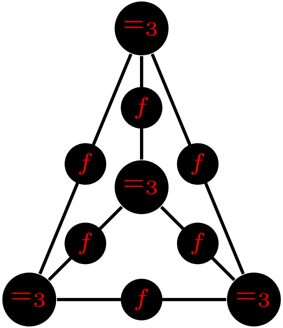
General Bipartite Holant Definition

- More generally, $\mathrm{Holant}(\mathcal{G} \,|\, \mathcal{R})$ is a counting problem defined over bipartite graphs.



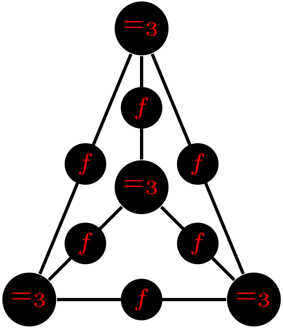
$$\sum_{\sigma : E \to \{0,1\}} \prod_{v \in V} f_v \left(\sigma \mid_{E(v)} \right)$$

Symmetric vs Asymmetric Function

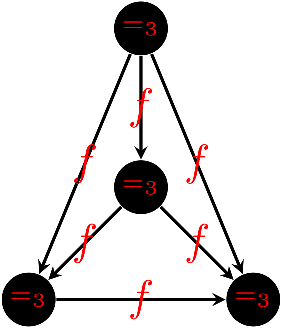


Input		Output
p	q	$f(p,q)$
0	0	w
0	1	x
1	0	y
1	1	z

Symmetric vs Asymmetric Function



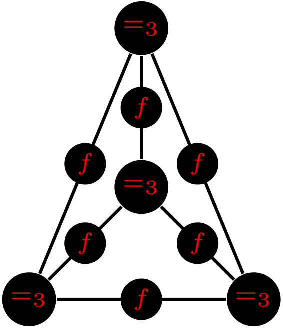
Input		Output
p	q	$f(p,q)$
0	0	w
0	1	x
1	0	y
1	1	z



- Define p to be on the tail
- Define q to be on the head

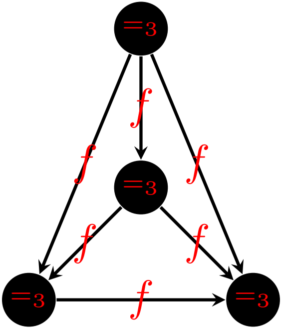
Symmetric vs Asymmetric Function

- (2,3)-regular



Input		Output
p	q	$f(p,q)$
0	0	w
0	1	x
1	0	y
1	1	z

- Directed 3-regular



- Define p to be on the tail
- Define q to be on the head

Strategy for Proving #P-hardness

- $\#\textsc{VertexCover}$ is #P-hard over 3-regular graphs.
- $\operatorname{Holant}(\{\mathrm{OR}_2\} \mid \{=_3\})$ is $\#\textsc{VertexCover}$ on 3-regular graphs.

Strategy for Proving $\#\mathrm{P}$-hardness

- $\#\textsc{VertexCover}$ is $\#\mathrm{P}$-hard over 3-regular graphs.
- $\mathrm{Holant}(\{\mathrm{OR}_2\} \mid \{=_3\})$ is $\#\textsc{VertexCover}$ on 3-regular graphs.

- Our problem is $\mathrm{Holant}(\{f\} \mid \{=_3\})$.
- Goal: simulate OR_2 using f.

Strategy for Proving $\#\mathrm{P}$-hardness

- $\#\textsc{VertexCover}$ is $\#\mathrm{P}$-hard over 3-regular graphs.
- $\mathrm{Holant}(\{\mathrm{OR}_2\} \mid \{=_3\})$ is $\#\textsc{VertexCover}$ on 3-regular graphs.

- Our problem is $\mathrm{Holant}(\{f\} \mid \{=_3\})$.
- Goal: simulate OR_2 using f.

- First step:

$$\mathrm{Holant}(\{\mathrm{OR}_2\} \mid \{=_3\}) \le_m^{\mathrm{P}} \mathrm{Holant}(\{f\} \cup \mathcal{U} \mid \{=_3\})$$

where $\mathcal{U}$ is the set of all unary functions.

Strategy for Proving $\#\mathrm{P}$-hardness

- $\#\textsc{VertexCover}$ is $\#\mathrm{P}$-hard over 3-regular graphs.
- $\mathrm{Holant}(\{\mathrm{OR}_2\} \mid \{=_3\})$ is $\#\textsc{VertexCover}$ on 3-regular graphs.

- Our problem is $\mathrm{Holant}(\{f\} \mid \{=_3\})$.
- Goal: simulate OR_2 using f.

- First step:

$$\mathrm{Holant}(\{\mathrm{OR}_2\} \mid \{=_3\}) \leq_m^{\mathrm{P}} \mathrm{Holant}(\{f\} \cup \mathcal{U} \mid \{=_3\})$$

where $\mathcal{U}$ is the set of all unary functions.

- Second step:

$$\mathrm{Holant}(\{f\} \cup \mathcal{U} \mid \{=_3\}) \leq_T^{\mathrm{P}} \mathrm{Holant}(\{f\} \mid \{=_3\})$$

Strategy for Proving $\#\mathrm{P}$-hardness

- $\#\textsc{VertexCover}$ is $\#\mathrm{P}$-hard over 3-regular graphs.
- $\operatorname{Holant}(\{\mathrm{OR}_2\} \mid \{=_3\})$ is $\#\textsc{VertexCover}$ on 3-regular graphs.

- Our problem is $\operatorname{Holant}(\{f\} \mid \{=_3\})$.
- Goal: simulate OR_2 using f.

- First step:

$$\operatorname{Holant}(\{\mathrm{OR}_2\} \mid \{=_3\}) \leq_m^{\mathrm{P}} \operatorname{Holant}(\{f\} \cup \mathcal{U} \mid \{=_3\})$$

  where $\mathcal{U}$ is the set of all unary functions.

- Second step:

$$\operatorname{Holant}(\{f\} \cup \mathcal{U} \mid \{=_3\}) \leq_T^{\mathrm{P}} \operatorname{Holant}(\{f\} \mid \{=_3\})$$

- Obtain $\mathcal{U}$ via interpolation.

Interpolation

- A degree n polynomial is uniquely defined by

Interpolation

- A degree n polynomial is uniquely defined by
    - $n + 1$ coefficients

Interpolation

- A degree n polynomial is uniquely defined by
  - $n + 1$ coefficients, or
  - evaluations at $n + 1$ (different) points.

Interpolation

- A degree n polynomial is uniquely defined by
  - $n+1$ coefficients, or
  - evaluations at $n+1$ (different) points.
- Interpolation is the process of converting from evaluations to coefficients.

Interpolation

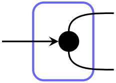
- A degree n polynomial is uniquely defined by
  - $n+1$ coefficients, or
  - evaluations at $n+1$ (different) points.
- Interpolation is the process of converting from evaluations to coefficients.
- We construct unary functions g_i such that the evaluation points are $\frac{g_i(0)}{g_i(1)}$.

Interpolation

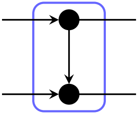
- A degree n polynomial is uniquely defined by
  - $n+1$ coefficients, or
  - evaluations at $n+1$ (different) points.
- Interpolation is the process of converting from evaluations to coefficients.
- We construct unary functions g_i such that the evaluation points are $\frac{g_i(0)}{g_i(1)}$.
- Distinct evaluation points $\iff$ unary functions pairwise linearly independent (as length-2 vectors).

Construction of Unary Functions

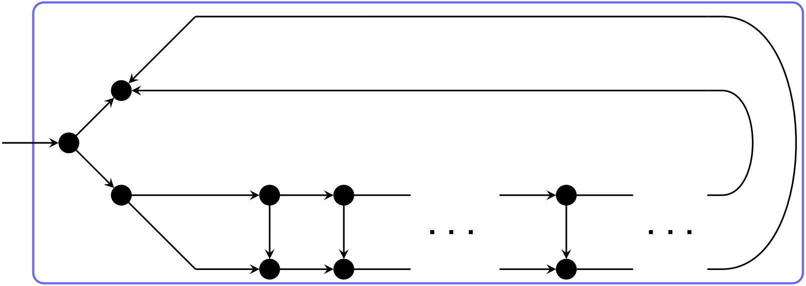
Projective Gadget



Recursive Gadget

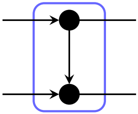


Unary Function



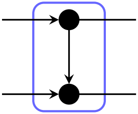
Matrix Representation

- Left side indexes the row.
- Right side indexes the column.
- High order bit on top.



Matrix Representation

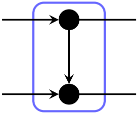
- Left side indexes the row.
- Right side indexes the column.
- High order bit on top.



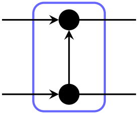
$$\begin{bmatrix} w & x \\ y & z \end{bmatrix}^{\otimes 2} \begin{bmatrix} w & 0 & 0 & 0 \\ 0 & x & 0 & 0 \\ 0 & 0 & y & 0 \\ 0 & 0 & 0 & z \end{bmatrix}$$

Matrix Representation

- Left side indexes the row.
- Right side indexes the column.
- High order bit on top.

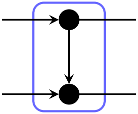


$$\begin{bmatrix} w & x \\ y & z \end{bmatrix}^{\otimes 2} \begin{bmatrix} w & 0 & 0 & 0 \\ 0 & x & 0 & 0 \\ 0 & 0 & y & 0 \\ 0 & 0 & 0 & z \end{bmatrix}$$

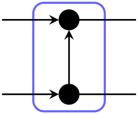


Matrix Representation

- Left side indexes the row.
- Right side indexes the column.
- High order bit on top.



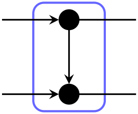
$$\begin{bmatrix} w & x \\ y & z \end{bmatrix}^{\otimes 2} \begin{bmatrix} w & 0 & 0 & 0 \\ 0 & x & 0 & 0 \\ 0 & 0 & y & 0 \\ 0 & 0 & 0 & z \end{bmatrix}$$



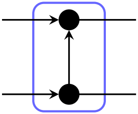
$$\begin{bmatrix} w & x \\ y & z \end{bmatrix}^{\otimes 2} \begin{bmatrix} w & 0 & 0 & 0 \\ 0 & y & 0 & 0 \\ 0 & 0 & x & 0 \\ 0 & 0 & 0 & z \end{bmatrix}$$

Matrix Representation

- Left side indexes the row.
- Right side indexes the column.
- High order bit on top.



$$\begin{bmatrix} w & x \\ y & z \end{bmatrix}^{\otimes 2} \begin{bmatrix} w & 0 & 0 & 0 \\ 0 & x & 0 & 0 \\ 0 & 0 & y & 0 \\ 0 & 0 & 0 & z \end{bmatrix}$$



$$\begin{bmatrix} w & x \\ y & z \end{bmatrix}^{\otimes 2} \begin{bmatrix} w & 0 & 0 & 0 \\ 0 & y & 0 & 0 \\ 0 & 0 & x & 0 \\ 0 & 0 & 0 & z \end{bmatrix}$$

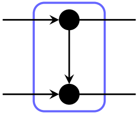
- Matrix of the composition is the product of the component matrices.

Anti-Gadget Construction

- Want set of matrix powers to form an infinite set of pairwise linearly independent matrices.

Anti-Gadget Construction

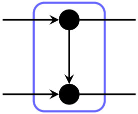
- Want set of matrix powers to form an infinite set of pairwise linearly independent matrices.
- If this matrix has this property, then we are done.



$$\begin{bmatrix} w & x \\ y & z \end{bmatrix}^{\otimes 2} \begin{bmatrix} w & 0 & 0 & 0 \\ 0 & x & 0 & 0 \\ 0 & 0 & y & 0 \\ 0 & 0 & 0 & z \end{bmatrix}$$

Anti-Gadget Construction

- Want set of matrix powers to form an infinite set of pairwise linearly independent matrices.
- If this matrix has this property, then we are done.

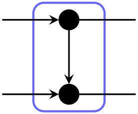


$$\begin{bmatrix} w & x \\ y & z \end{bmatrix}^{\otimes 2} \begin{bmatrix} w & 0 & 0 & 0 \\ 0 & x & 0 & 0 \\ 0 & 0 & y & 0 \\ 0 & 0 & 0 & z \end{bmatrix}$$

- Otherwise, some power k is a multiple of the identity matrix.

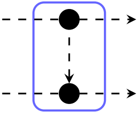
Anti-Gadget Construction

- Want set of matrix powers to form an infinite set of pairwise linearly independent matrices.
- If this matrix has this property, then we are done.



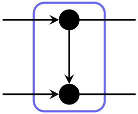
$$\begin{bmatrix} w & x \\ y & z \end{bmatrix}^{\otimes 2} \begin{bmatrix} w & 0 & 0 & 0 \\ 0 & x & 0 & 0 \\ 0 & 0 & y & 0 \\ 0 & 0 & 0 & z \end{bmatrix}$$

- Otherwise, some power k is a multiple of the identity matrix.
- Using only $k-1$ compositions creates an anti-gadget.



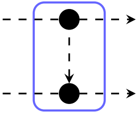
Anti-Gadget Construction

- Want set of matrix powers to form an infinite set of pairwise linearly independent matrices.
- If this matrix has this property, then we are done.



$$\begin{bmatrix} w & x \\ y & z \end{bmatrix}^{\otimes 2} \begin{bmatrix} w & 0 & 0 & 0 \\ 0 & x & 0 & 0 \\ 0 & 0 & y & 0 \\ 0 & 0 & 0 & z \end{bmatrix}$$

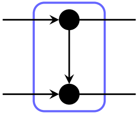
- Otherwise, some power k is a multiple of the identity matrix.
- Using only $k-1$ compositions creates an anti-gadget.



$$\left(\begin{bmatrix} w & x \\ y & z \end{bmatrix}^{\otimes 2} \begin{bmatrix} w & 0 & 0 & 0 \\ 0 & x & 0 & 0 \\ 0 & 0 & y & 0 \\ 0 & 0 & 0 & z \end{bmatrix} \right)^{-1}$$

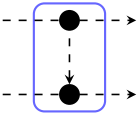
Anti-Gadget Construction

- Want set of matrix powers to form an infinite set of pairwise linearly independent matrices.
- If this matrix has this property, then we are done.



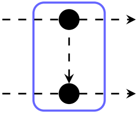
$$\begin{bmatrix} w & x \\ y & z \end{bmatrix}^{\otimes 2} \begin{bmatrix} w & 0 & 0 & 0 \\ 0 & x & 0 & 0 \\ 0 & 0 & y & 0 \\ 0 & 0 & 0 & z \end{bmatrix}$$

- Otherwise, some power k is a multiple of the identity matrix.
- Using only $k-1$ compositions creates an anti-gadget.



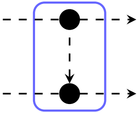
$$\left(\begin{bmatrix} w & 0 & 0 & 0 \\ 0 & x & 0 & 0 \\ 0 & 0 & y & 0 \\ 0 & 0 & 0 & z \end{bmatrix} \right)^{-1} \left(\begin{bmatrix} w & x \\ y & z \end{bmatrix}^{\otimes 2} \right)^{-1}$$

Anti-Gadget Technique

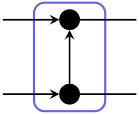


$$\left(\begin{bmatrix} w & 0 & 0 & 0 \\ 0 & x & 0 & 0 \\ 0 & 0 & y & 0 \\ 0 & 0 & 0 & z \end{bmatrix}\right)^{-1} \left(\begin{bmatrix} w & x \\ y & z \end{bmatrix}^{\otimes 2}\right)^{-1}$$

Anti-Gadget Technique

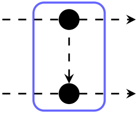


$$\left(\begin{bmatrix} w & 0 & 0 & 0 \\ 0 & x & 0 & 0 \\ 0 & 0 & y & 0 \\ 0 & 0 & 0 & z \end{bmatrix}\right)^{-1} \left(\begin{bmatrix} w & x \\ y & z \end{bmatrix}^{\otimes 2}\right)^{-1}$$

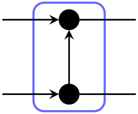


$$\begin{bmatrix} w & x \\ y & z \end{bmatrix}^{\otimes 2} \begin{bmatrix} w & 0 & 0 & 0 \\ 0 & y & 0 & 0 \\ 0 & 0 & x & 0 \\ 0 & 0 & 0 & z \end{bmatrix}$$

Anti-Gadget Technique



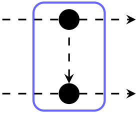
$$\left(\begin{bmatrix} w & 0 & 0 & 0 \\ 0 & {\color{red}x} & 0 & 0 \\ 0 & 0 & {\color{red}y} & 0 \\ 0 & 0 & 0 & z \end{bmatrix}\right)^{-1} \left(\begin{bmatrix} w & x \\ y & z \end{bmatrix}^{\otimes 2}\right)^{-1}$$



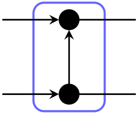
$$\begin{bmatrix} w & x \\ y & z \end{bmatrix}^{\otimes 2} \begin{bmatrix} w & 0 & 0 & 0 \\ 0 & {\color{red}y} & 0 & 0 \\ 0 & 0 & {\color{red}x} & 0 \\ 0 & 0 & 0 & z \end{bmatrix}$$

- The composition of these two gadgets yields...

Anti-Gadget Technique

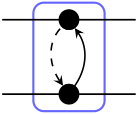


$$\left(\begin{bmatrix} w & 0 & 0 & 0 \\ 0 & x & 0 & 0 \\ 0 & 0 & y & 0 \\ 0 & 0 & 0 & z \end{bmatrix}\right)^{-1} \quad \left(\begin{bmatrix} w & x \\ y & z \end{bmatrix}^{\otimes 2}\right)^{-1}$$



$$\begin{bmatrix} w & x \\ y & z \end{bmatrix}^{\otimes 2} \begin{bmatrix} w & 0 & 0 & 0 \\ 0 & y & 0 & 0 \\ 0 & 0 & x & 0 \\ 0 & 0 & 0 & z \end{bmatrix}$$

- The composition of these two gadgets yields...



$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{y}{x} & 0 & 0 \\ 0 & 0 & \frac{x}{y} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The First Anti-Gadget Lemma

Lemma

For $w, x, y, z \in \mathbb{C}$, *if*

- $wz \neq xy$,
- $wxyz \neq 0$, *and*
- $|x| \neq |y|$,

then there exists a recursive gadget whose matrix powers form an infinite set of pairwise linearly independent matrices.

The First Anti-Gadget Lemma

Lemma

For $w, x, y, z \in \mathbb{C}$, *if*

- $wz \neq xy$,
- $wxyz \neq 0$, *and*
- $|x| \neq |y|$,

then there exists a recursive gadget whose matrix powers form an infinite set of pairwise linearly independent matrices.

Corollary

For $w, x, y, z \in \mathbb{C}$ *as above,* $\text{Holant}(\{f\} \mid \{=_3\})$ *is* $\#\text{P}$*-hard.*

Thank You

Thank You

Paper and slides available on my website.
www.cs.wisc.edu/~tdw