Reductions and completeness

- $\#3\textsc{Sat}$ is #P-complete under parsimonious reductions (Exercise).

Reductions and completeness

- $\#3\textsc{Sat}$ is #P-complete under parsimonious reductions (Exercise).
- $\textsc{Permanent}$ is #P-complete under Turing reductions (Not an exercise).

Reductions and completeness

- #3SAT is #P-complete under parsimonious reductions (Exercise).
- PERMANENT is #P-complete under Turing reductions (Not an exercise).
  - If PERMANENT is #P-complete under parsimonious reductions, then P = NP (Exercise).

Reductions and completeness

- #3SAT is #P-complete under parsimonious reductions (Exercise).
- PERMANENT is #P-complete under Turing reductions (Not an exercise).
  - If PERMANENT is #P-complete under parsimonious reductions, then P = NP (Exercise).
- Every #P-complete problem under parsimonious reductions has an NP-complete decision version (Execise).

Reductions and completeness

- #3Sat is #P-complete under parsimonious reductions (Exercise).
- Permanent is #P-complete under Turing reductions
(Not an exercise).
  - If Permanent is #P-complete under parsimonious reductions, then $P = NP$ (Exercise).
- Every #P-complete problem under parsimonious reductions has an NP-complete decision version (Execise).
- Conjecture: Every NP-complete problem has a #P-complete counting version.

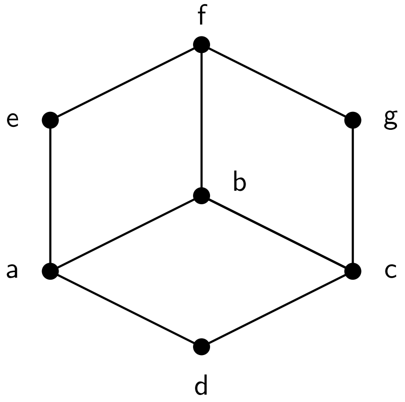
Some basic inclusions

- FP $\subseteq$ #P $\subseteq$ FPSPACE.
- NP $\subseteq$ $\mathrm{P}^{\#\mathrm{P}[1]}$.
- If FP $=$ #P, then P $=$ NP.
- **Toda's Theorem:** PH $\subseteq$ $\mathrm{P}^{\#\mathrm{P}[1]}$.

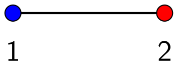
Overview

Graph Homomorphism Problems

Input graph **G**

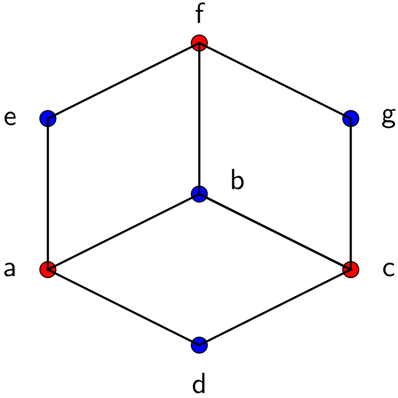


Target graph **H**

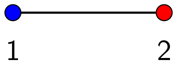


Graph Homomorphism Problems

Input graph **G**



Target graph **H**

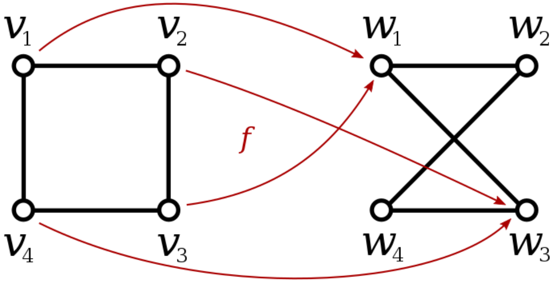


- Every homomorphism from G to H is a 2-coloring of G.
- The number of homomorphisms from G to H is equal to $\#2\text{-}\textsc{Colorings}(G)$.

Graph homomorphisms preserve structure

Definition

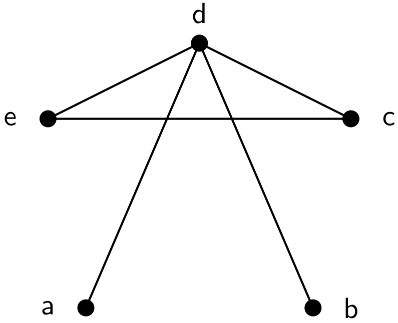
Given graphs G and H, a homomorphism from G to H is a function $f : V(G) \to V(H)$ such that every edge $(u, v) \in E(G)$ is mapped to an edge $(f(u), f(v)) \in E(H)$.

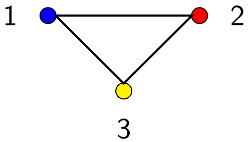


Counting graph homomorphisms or H-colorings of G

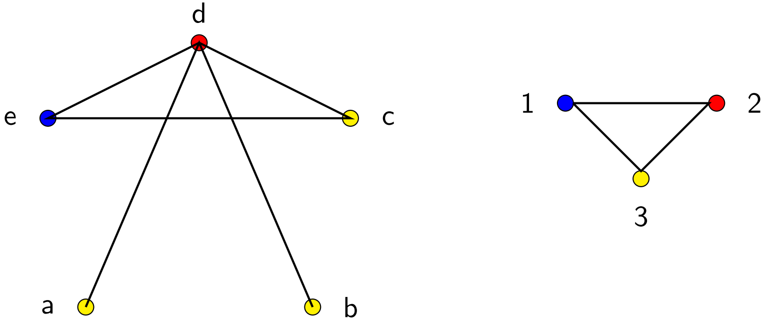
- We denote by $\mathrm{Hom}(G, H)$ the number of homomorphisms from G to H.
- Graph homomorphisms from G to H are also called H-colorings of G.
- We denote by #HomsToH (or #H-Colorings) the problem of counting the number of homomorphisms from an input graph to a fixed graph H.

Other examples – #3-COLORINGS



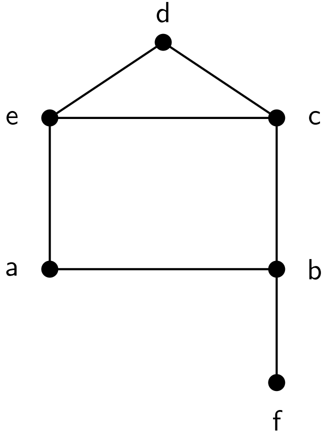


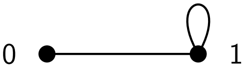
Other examples – #3-Colorings



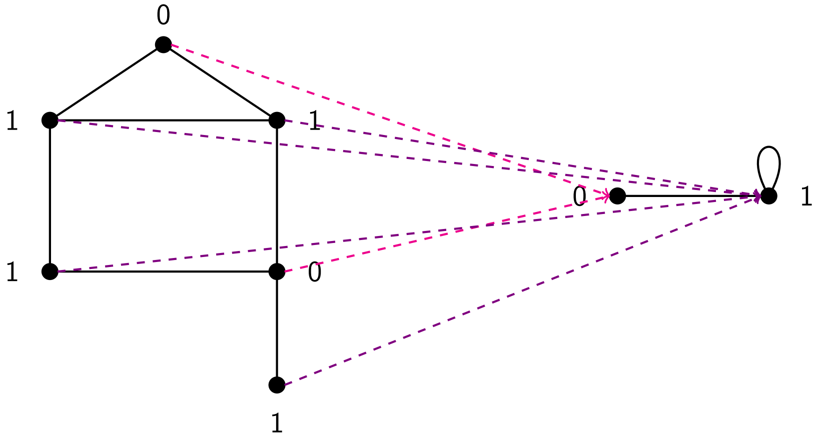
- $\text{Hom}(G, K_3) = \#3\text{-Colorings}(G)$.
- $\text{Hom}(G, K_q) = \#q\text{-Colorings}(G)$ for any $q \geq 2$.

Other examples – #VERTEXCOVERS

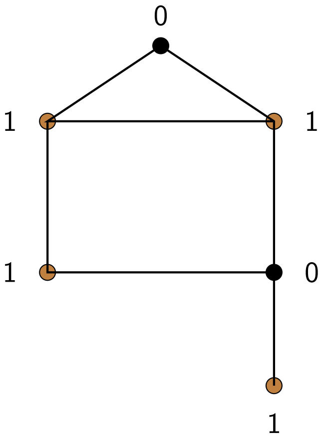




Other examples – #VertexCovers

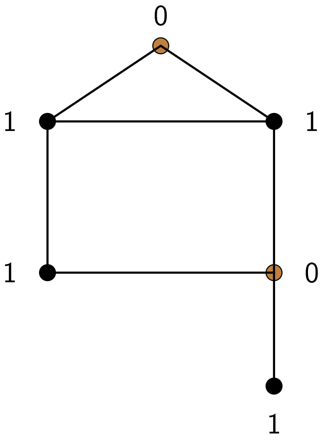


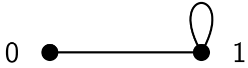
Other examples – #VertexCovers



- The subset of vertices mapped to 1 form a vertex cover.
- $\#\textsc{HomsToH}(G) = \#\textsc{VertexCovers}(G)$.

Other examples – #IS





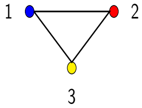
- The subset of vertices mapped to 0 form an independent set.
- $\#\textsc{HomsToH}(G) = \#\text{IS}(G)$.

- Let A be the (0-1) adjacency matrix of H.
- Given a graph G, $\#\text{HomsToH}$ is the problem of computing the following sum

$$Z_A(G) = \sum_{\sigma:V(G)\to V(H)} \prod_{(u,v)\in E(G)} A(\sigma(u),\sigma(v)).$$

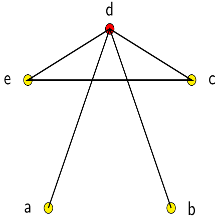
- Sometimes we write $Z_H(G)$ instead of $Z_A(G)$.

For example,



has $A = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}$.

So, the following $\sigma : V \rightarrow [3]$



contributes 0, since
$A(\sigma(e), \sigma(c)) = A(3,3) = 0$.

Weighted graph homomorphisms

- In general, H can be a weighted graph, and so A can be a matrix over the real or complex numbers.

Weighted graph homomorphisms

- In general, H can be a weighted graph, and so A can be a matrix over the real or complex numbers.

- The product $\prod_{(u,v) \in E(G)} A(\sigma(u), \sigma(v))$ is the weight of assignment σ.

Weighted graph homomorphisms

- In general, H can be a weighted graph, and so A can be a matrix over the real or complex numbers.
- The product $\prod_{(u,v)\in E(G)} A(\sigma(u), \sigma(v))$ is the weight of assignment σ.
- The sum over all assignments

$$Z_A(G) = \sum_{\sigma:V(G)\to V(H)} \prod_{(u,v)\in E(G)} A(\sigma(u), \sigma(v))$$

is called the partition function.

Weighted graph homomorphisms

- Motivation and applications come from statistical physics.
- To capture the computational complexity of such functions, we consider the closure of $\#\mathsf{P}$ under Turing reductions.

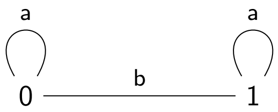
  That is the class $\mathsf{FP}^{\#\mathsf{P}}$.

Partition functions in statistical physics

- In statistical physics, partition functions can be represented as weighted graph homomorphisms.
- In this case, H is a weighted graph with both edge and vertex weights.
- Let A be the adjacency matrix of H, where $A(u, v)$ is the weight of the edge $(u, v) \in E(H)$ and let $\{\lambda_v\}_{v \in V(H)}$ be the vertex weights.
- Weighted $\#\text{HomsToH}$ is the problem of computing the following sum

$$\sum_{\sigma : V(G) \to V(H)} \Big(\prod_{(u,v) \in E(G)} A(\sigma(u), \sigma(v)) \prod_{v \in V(G)} \lambda_{\sigma(v)} \Big).$$

2-spin systems – Ising model



Adjacency matrix

$$A = \begin{bmatrix} a & b \\ b & a \end{bmatrix}$$

Vertex weights

$\lambda_0 = 1$ and $\lambda_1 = \lambda$

$$Z_{a,b,\lambda}(G) =$$

$$\sum_{\sigma:V\to\{0,1\}} a^{|\{(u,v)\in E:\sigma(u)=\sigma(v)\}|} \cdot b^{|\{(u,v)\in E:\sigma(u)\neq\sigma(v)\}|} \cdot \lambda^{|\{u\in V:\sigma(u)=1\}|}$$

2-spin systems – Hardcore model

Adjacency matrix

$$A = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}$$

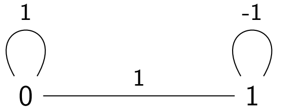
Vertex weights

$\lambda_0 = 1$ and $\lambda_1 = \lambda$

$$Z_\lambda(G) = \sum_{I \in \mathcal{I}} \lambda^{|I|}$$

where $\mathcal{I}$ is the set of all independent sets in G

Counting induced subgraphs with an even number of edges



Adjacency matrix

$$A = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

- The term $\prod_{(u,v)\in E} A(\sigma(u), \sigma(v))$ is 1 if the subgraph of G induced by vertices mapped to 1 has an even number of edges and it is -1, otherwise.
- $Z_A(G) = X - Y$, where X (resp. Y) is the number of induced subgraphs of G with an even (resp. odd) number of edges.
- $X + Y = 2^n$
- So,

$$X = \frac{2^n + Z_A(G)}{2}.$$

Constraint Satisfaction Problems

3SAT as a CSP:

- Variables $x_1, \ldots, x_n$
- Domain $\{0, 1\}$
- Constraints $(C_i, x_{i_1}, x_{i_2}, x_{i_3})$

Clause	Relation
$x_1 \vee x_2 \vee x_3$	$C_0 = \{0,1\}^3 \setminus (0,0,0)$
$\neg x_1 \vee x_2 \vee x_3$	$C_1 = \{0,1\}^3 \setminus (1,0,0)$
$\neg x_1 \vee \neg x_2 \vee x_3$	$C_2 = \{0,1\}^3 \setminus (1,1,0)$
$\neg x_1 \vee \neg x_2 \vee \neg x_3$	$C_3 = \{0,1\}^3 \setminus (1,1,1)$

Constraint Satisfaction Problems

Decision version

Input:

- A set of variables $x_1, ..., x_n$.
- A domain $[q]$ of size q.
- A set C of constraints $(C_i, x_{i_1}, ..., x_{i_k})$, where C_i are relations on the domain of arity k.

Output: Is there an assignment of values to the variables such that all constraints are satisfied?

Counting Constraint Satisfaction Problems

- A counting constraint satisfaction problem is parameterized by a set of local constraint functions $\mathcal{F}$.

Counting Constraint Satisfaction Problems

- A counting constraint satisfaction problem is parameterized by a set of local constraint functions $\mathcal{F}$.
- It is denoted by $\#\mathrm{CSP}_q(\mathcal{F})$ when the constraint functions in $\mathcal{F}$ are defined over a domain $[q]$ of size q.

Counting Constraint Satisfaction Problems

- A counting constraint satisfaction problem is parameterized by a set of local constraint functions $\mathcal{F}$.

- It is denoted by $\#\mathrm{CSP}_q(\mathcal{F})$ when the constraint functions in $\mathcal{F}$ are defined over a domain $[q]$ of size q.

- Every constraint is a function $f \in \mathcal{F}$ of some arity k together with a sequence of k variables $x_{i_1}, \ldots, x_{i_k} \in \{x_1, \ldots, x_n\}$.

Counting Constraint Satisfaction Problems

- A counting constraint satisfaction problem is parameterized by a set of local constraint functions $\mathcal{F}$.
- It is denoted by $\#\mathrm{CSP}_q(\mathcal{F})$ when the constraint functions in $\mathcal{F}$ are defined over a domain $[q]$ of size q.
- Every constraint is a function $f \in \mathcal{F}$ of some arity k together with a sequence of k variables $x_{i_1}, \ldots, x_{i_k} \in \{x_1, \ldots, x_n\}$.
- *Output:* How many assignments of values to the variables are there such that all constraints are satisfied?
  Equivalently, compute the following sum

$$\sum_{x_1,\ldots,x_n \in [q]} \prod_{(f, x_{i_1}, \ldots, x_{i_k}) \in C} f(x_{i_1}, \ldots, x_{i_k})$$

Examples of $\#\mathrm{CSP}_q(\mathcal{F})$ – $\#\textsc{Sat}$

- Variables $x_1, ..., x_n$
- Domain $\{0, 1\}$
- $\mathcal{F} = \{OR_k \mid k \geq 1\} \cup \{\neq_2\}$, where

$$OR_k(x_1, ..., x_k) = \begin{cases} 0, & \text{if } x_1 = ... = x_k = 0 \\ 1, & \text{otherwise} \end{cases} \quad \text{and}$$

$$\neq_2 (x_1, x_2) = \begin{cases} 0, & \text{if } x_1 = x_2 \\ 1, & \text{otherwise} \end{cases} .$$

Examples of $\#\mathrm{CSP}_2(\mathcal{F})$ – $\#\mathrm{SAT}$

For example, $(x_1 \vee x_2 \vee x_3) \wedge (\neg x_3 \vee x_4)$ corresponds to the following instance of $\#\mathrm{CSP}_2(\mathcal{F})$:

1. Variables x_1, x_2, x_3, x_4, x_5.
2. Constraints (OR_3, x_1, x_2, x_3), (OR_2, x_5, x_4), $(\neq_2, x_3, x_5)$.

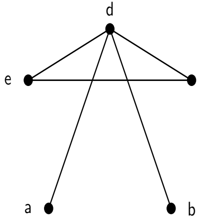
Examples of $\#\text{CSP}_2(\mathcal{F})$ – Counting satisfying assignments

Problem	Constraint functions
#3SAT	$\mathcal{F} = \{OR_3, \neq_2\}$
#NAE-3SAT	$\mathcal{F} = \{\textit{Not-All-Equal}_3, \neq_2\}$
#MONSAT	$\mathcal{F} = \{OR_k \mid k \geq 1\}$
#MON3SAT	$\mathcal{F} = \{OR_3\}$
#MON1-IN-3SAT	$\mathcal{F} = \{\textit{Exact-One}_3\}$

An Example of $\#\text{CSP}_3(\mathcal{F})$ – $\#$3-COLORINGS

- Variables $x_1, ..., x_n$
- Domain $\{1, 2, 3\}$
- $\mathcal{F} = \{\neq_2\}$

For example,



corresponds to the following instance of $\#\text{CSP}_3(\mathcal{F})$:

1. Variables x_a, x_b, x_c, x_d, x_e.
2. Constraints $(\neq_2, x_a, x_d)$, $(\neq_2, x_b, x_d)$, $(\neq_2, x_d, x_e)$, $(\neq_2, x_c, x_d)$, $(\neq_2, x_c, x_e)$.

Computing #HOMSTOH is a #CSP$_q(\mathcal{F})$

For graphs G and H, we construct a CSP instance with only one kind of constraint, as follows.

- The variables are the vertices of G.
- The domain is the vertex set of H.
- The constraints are $((u, v), E(H))$ for every edge $(u, v) \in E(G)$.

An assignment σ of values to the variables is a function from $V(G)$ to $V(H)$ such that $(\sigma(u), \sigma(v)) \in E(H)$ for every $(u, v) \in E(G)$.

Computing #HomsToH is a $\#\text{CSP}_q(\mathcal{F})$

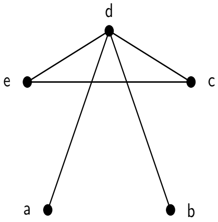
For graphs G and H, we construct a CSP instance with only one kind of constraint, as follows.

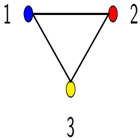
- The variables are the vertices of G.
- The domain is the vertex set of H.
- The constraints are $((u,v), E(H))$ for every edge $(u,v) \in E(G)$.

An assignment σ of values to the variables is a function from $V(G)$ to $V(H)$ such that $(\sigma(u), \sigma(v)) \in E(H)$ for every $(u,v) \in E(G)$.

For the counting version, we consider the corresponding constraint function $E_H : V(H) \times V(H) \to \{0,1\}$.

#3-Colorings revisited





1. Variables x_a, x_b, x_c, x_d, x_e.
2. Domain $\{1, 2, 3\}$
3. Constraints (E_H, x_a, x_d), (E_H, x_b, x_d), (E_H, x_d, x_e), (E_H, x_c, x_d), (E_H, x_c, x_e),

   where $E_H(x, y) = \begin{cases} 1, & \text{if } (x, y) \in E(H) \\ 0, & \text{otherwise} \end{cases}$.

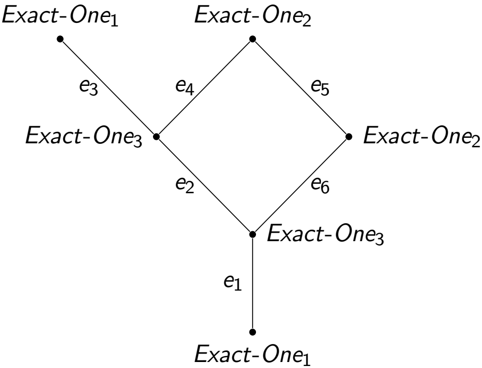
- $\#\mathrm{CSP}_q(\mathcal{F})$ is a generalization of counting graph homomorphisms.

- $\#\mathrm{CSP}_q(\mathcal{F})$ is a generalization of counting graph homomorphisms.
- Every $\#\mathrm{CSP}_q(\mathcal{F})$ is a problem of counting homomorphisms between two relational structures $\mathcal{G}$ and $\mathcal{H}$.

- $\#\mathrm{CSP}_q(\mathcal{F})$ is a generalization of counting graph homomorphisms.
- Every $\#\mathrm{CSP}_q(\mathcal{F})$ is a problem of counting homomorphisms between two relational structures $\mathcal{G}$ and $\mathcal{H}$.
- Weighted counting CSP are defined by considering constraint functions over the rational, real or complex numbers.

- $\#\mathrm{CSP}_q(\mathcal{F})$ is a generalization of counting graph homomorphisms.
- Every $\#\mathrm{CSP}_q(\mathcal{F})$ is a problem of counting homomorphisms between two relational structures $\mathcal{G}$ and $\mathcal{H}$.
- Weighted counting CSP are defined by considering constraint functions over the rational, real or complex numbers.
- Boolean CSP are CSP with Boolean domain $\{0, 1\}$.

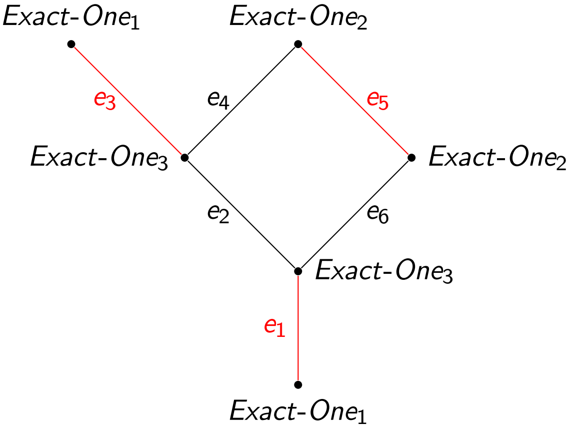
Holant problems



$$Exact\text{-}One_2(e_i, e_j) = \begin{cases} 1, & \text{if } (e_i, e_j) = (0,1) \text{ or } (e_i, e_j) = (1,0) \\ 0, & \text{otherwise} \end{cases}$$

The subscript k in $Exact\text{-}One_k$ denotes the arity of the function.

Holant problems



Holant problems

- A Holant problem is parameterized by a set $\mathcal{F}$ of local constraint functions, also called signatures.

Holant problems

- A Holant problem is parameterized by a set $\mathcal{F}$ of local constraint functions, also called signatures.
- A signature grid $\Omega = (G, \pi)$ over $\mathcal{F}$ consists of a graph $G = (V, E)$ and a mapping π that assigns to each vertex $u \in V$ an $f_u \in \mathcal{F}$ and a linear order on the incident edges at u.

Holant problems

- A Holant problem is parameterized by a set $\mathcal{F}$ of local constraint functions, also called signatures.
- A signature grid $\Omega = (G, \pi)$ over $\mathcal{F}$ consists of a graph $G = (V, E)$ and a mapping π that assigns to each vertex $u \in V$ an $f_u \in \mathcal{F}$ and a linear order on the incident edges at u.
- The arity of f_u is equal to the degree of u, and the incident edges at u are associated with the input variables of f_u.

Holant problems

- A Holant problem is parameterized by a set $\mathcal{F}$ of local constraint functions, also called signatures.
- A signature grid $\Omega = (G, \pi)$ over $\mathcal{F}$ consists of a graph $G = (V, E)$ and a mapping π that assigns to each vertex $u \in V$ an $f_u \in \mathcal{F}$ and a linear order on the incident edges at u.
- The arity of f_u is equal to the degree of u, and the incident edges at u are associated with the input variables of f_u.
- A signature is symmetric if its value is invariant under permutation of its variables.

Holant problems

- A Holant problem is parameterized by a set $\mathcal{F}$ of local constraint functions, also called signatures.
- A signature grid $\Omega = (G, \pi)$ over $\mathcal{F}$ consists of a graph $G = (V, E)$ and a mapping π that assigns to each vertex $u \in V$ an $f_u \in \mathcal{F}$ and a linear order on the incident edges at u.
- The arity of f_u is equal to the degree of u, and the incident edges at u are associated with the input variables of f_u.
- A signature is symmetric if its value is invariant under permutation of its variables.
- If all signatures in $\mathcal{F}$ are symmetric, then there is no need to assign an order for incident edges at any u.

Holant problems

Definition

For a set $\mathcal{F}$ of signatures over a domain $[q]$, we define $\text{Holant}_q(\mathcal{F})$ as the problem with

Input: A signature grid $\Omega = (G, \pi)$ over $\mathcal{F}$

Output:

$$\text{Holant}_q(\Omega, \mathcal{F}) = \sum_{\sigma : E \to [q]} \prod_{u \in V} f_u(\sigma \restriction_{E(u)})$$

- $G = (V, E)$ and $E(u)$ denotes the incident edges of u.
- $\sigma \restriction_{E(u)}$ denotes the restriction of σ to $E(u)$.
- $f_u(\sigma \restriction_{E(u)})$ is the evaluation of f_u on the ordered input tuple $\sigma \restriction_{E(u)}$.
- We use $\text{Holant}(\mathcal{F})$ to denote $\text{Holant}_2(\mathcal{F})$, the Holant problems over the Boolean domain.

Some notation

- A signature f of arity k over the Boolean domain can be denoted by $(f_0, f_1, \ldots, f_{2^k-1})$, where f_x is the output of f on $x \in [2]^k$, ordered lexicographically.

Some notation

- A signature f of arity k over the Boolean domain can be denoted by $(f_0, f_1, ..., f_{2^k-1})$, where f_x is the output of f on $x \in [2]^k$, ordered lexicographically.
- A signature f of arity k over the Boolean domain is symmetric if for every $x, y \in \{0,1\}^k$ of equal Hamming weight, $f(x) = f(y)$.
- In this case, f can also be expressed as $[f_0, f_1, ..., f_k]$, where f_w is the value of f on inputs of Hamming weight w.

Some notation

- A signature f of arity k over the Boolean domain can be denoted by $(f_0, f_1, ..., f_{2^k-1})$, where f_x is the output of f on $x \in [2]^k$, ordered lexicographically.
- A signature f of arity k over the Boolean domain is symmetric if for every $x, y \in \{0,1\}^k$ of equal Hamming weight, $f(x) = f(y)$.
- In this case, f can also be expressed as $[f_0, f_1, ..., f_k]$, where f_w is the value of f on inputs of Hamming weight w.
- For example, *Equality* signature of arity k,

$$(=_k) = [1, 0, ..., 0, 1]$$

and *Disequality* signature of arity 2,

$$(\neq_2) = [0, 1, 0].$$

Holant problems – Examples

Examples of counting problems in k-regular graphs.

- Over the Boolean domain:

$$\text{Holant}(G; f) \text{ counts } \begin{cases} \text{matchings} & \text{in } G \text{ when } f = \textsc{At-Most-One}_k; \\ \text{perfect matchings} & \text{in } G \text{ when } f = \textsc{Exact-One}_k; \\ \text{cycle covers} & \text{in } G \text{ when } f = \textsc{Exact-Two}_k; \\ \text{edge covers} & \text{in } G \text{ when } f = \textsc{Or}_k. \end{cases}$$

- Over domain of size q, $\text{Holant}_q(G, \textit{All-Distinct}_k)$ is the problem $\#q$-$\textsc{EdgeColorings}$.

Planar / bipartite signature grids

- A planar signature grid is a signature grid s.t. its underlying graph is planar. We use PI-$\text{Holant}_q(\mathcal{F})$ to denote the restriction of $\text{Holant}_q(\mathcal{F})$ to planar signature grids.

Planar / bipartite signature grids

- A planar signature grid is a signature grid s.t. its underlying graph is planar. We use $\text{Pl-Holant}_q(\mathcal{F})$ to denote the restriction of $\text{Holant}_q(\mathcal{F})$ to planar signature grids.
- A bipartite signature grid over $(\mathcal{F} \mid \mathcal{G})$ is a signature grid $\Omega = (H, \pi)$ over $\mathcal{F} \cup \mathcal{G}$, where $H = (V_1 \cup V_2, E)$ is a bipartite graph, s.t. $\pi(V_1) \subseteq \mathcal{F}$ and $\pi(V_2) \subseteq \mathcal{G}$. We use $\text{Holant}_q(\mathcal{F} \mid \mathcal{G})$ to denote the restriction of $\text{Holant}_q(\mathcal{F} \cup \mathcal{G})$ to bipartite signatures over $(\mathcal{F} \mid \mathcal{G})$.