Query Algorithms Based on Homomorphism Counts

Wei-Lin Wu
University of California Santa Cruz, USA

ICALP Structure Meets Power Workshop

July 04, 2022

Joint work with Phokion G. Kolaitis

Homomorphism Counts

Let G and H be two graphs (**finite**, **undirected** and **simple**).

1. Homomorphism from G to H: A function $h : V(G) \to V(H)$ such that for all $u, v \in V(G)$:
$$\text{if} \quad (u, v) \in E(G), \quad \text{then} \quad (h(u), h(v)) \in E(H).$$

Homomorphism Counts

Let G and H be two graphs (**finite**, **undirected** and **simple**).

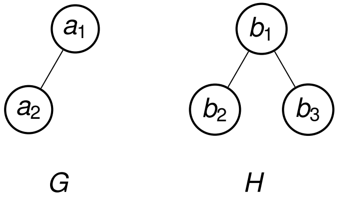
1. Homomorphism from G to H: A function $h : V(G) \rightarrow V(H)$ such that for all $u, v \in V(G)$:
   $$\text{if} \quad (u, v) \in E(G), \quad \text{then} \quad (h(u), h(v)) \in E(H).$$
2. $\hom(G, H)$: **number** of homomorphisms from G to H.

Homomorphism Counts

Let G and H be two graphs (**finite**, **undirected** and **simple**).

1. Homomorphism from G to H: A function $h : V(G) \to V(H)$ such that for all $u, v \in V(G)$:
$$\text{if} \quad (u, v) \in E(G), \quad \text{then} \quad (h(u), h(v)) \in E(H).$$
2. $\text{hom}(G, H)$: **number** of homomorphisms from G to H.

E.g., $\text{hom}(G, H) = 4$ for the graphs G and H below:



Query Algorithms Based on Homomorphism Counts

Let $\mathcal{C}$ be a **class** of (isomorphism types of) graphs.

Query Algorithms Based on Homomorphism Counts

Let $\mathcal{C}$ be a **class** of (isomorphism types of) graphs.

Chen, Flum, Liu and Xun (2022) introduced the notions below:

1. $\mathcal{C}$ admits a k-non-adaptive left algorithm ($k \geq 1$) if there are k fixed graphs $F_1, F_2, \ldots, F_k$ such that for every G,
$$G \in \mathcal{C} \quad \text{iff} \quad (n_1, n_2, \ldots, n_k) \in X$$
where
   - $X \subseteq \mathbb{N}^k$ is decidable ($\mathbb{N}$: non-negative integers), and
   - $n_1 := \hom(F_1, G), n_2 := \hom(F_2, G), \ldots, n_k := \hom(F_k, G)$ are left homomorphism counts as queries made to the input G.

Query Algorithms Based on Homomorphism Counts

Let $\mathcal{C}$ be a **class** of (isomorphism types of) graphs.

Chen, Flum, Liu and Xun (2022) introduced the notions below:

1. $\mathcal{C}$ admits a k-non-adaptive left algorithm ($k \geq 1$) if there are k fixed graphs $F_1, F_2, \ldots, F_k$ such that for every G,
$$G \in \mathcal{C} \quad \text{iff} \quad (n_1, n_2, \ldots, n_k) \in X$$
where
   - $X \subseteq \mathbb{N}^k$ is decidable ($\mathbb{N}$: non-negative integers), and
   - $n_1 := \hom(F_1, G), n_2 := \hom(F_2, G), \ldots, n_k := \hom(F_k, G)$ are left homomorphism counts as queries made to the input G.
2. $\mathcal{C}$ admits a k-adaptive left algorithm ($k \geq 1$) if the same holds except that $F_i = F_i(n_1, \ldots, n_{i-1})$ is a function of $n_1, \ldots, n_{i-1}$ for $2 \leq i \leq k$.

Query Algorithms Based on Homomorphism Counts

Let $\mathcal{C}$ be a **class** of (isomorphism types of) graphs.

Chen, Flum, Liu and Xun (2022) introduced the notions below:

1. $\mathcal{C}$ admits a k-non-adaptive left algorithm ($k \geq 1$) if there are k fixed graphs $F_1, F_2, \ldots, F_k$ such that for every G,
$$G \in \mathcal{C} \quad \text{iff} \quad (n_1, n_2, \ldots, n_k) \in X$$
where
   - $X \subseteq \mathbb{N}^k$ is decidable ($\mathbb{N}$: non-negative integers), and
   - $n_1 := \hom(F_1, G), n_2 := \hom(F_2, G), \ldots, n_k := \hom(F_k, G)$ are left homomorphism counts as queries made to the input G.
2. $\mathcal{C}$ admits a k-adaptive left algorithm ($k \geq 1$) if the same holds except that $F_i = F_i(n_1, \ldots, n_{i-1})$ is a function of $n_1, \ldots, n_{i-1}$ for $2 \leq i \leq k$.
3. "$\mathcal{C}$ admits a k-(non-)adaptive right algorithm" is analogous.

Non-Adaptive Left Algorithms

Theorem (Chen-Flum-Liu-Xun 2022):

The following classes admit a non-adaptive left algorithm (for some $k \geq 1$):

1. class of graphs definable by a Boolean combination of universal first-order sentences

Non-Adaptive Left Algorithms

Theorem (Chen-Flum-Liu-Xun 2022):

The following classes admit a non-adaptive left algorithm (for some $k \geq 1$):

1. class of graphs definable by a Boolean combination of universal first-order sentences
2. class of 3-regular (or any m-regular) graphs

Non-Adaptive Left Algorithms

Theorem (Chen-Flum-Liu-Xun 2022):

The following classes admit a non-adaptive left algorithm (for some $k \geq 1$):

1. class of graphs definable by a Boolean combination of universal first-order sentences
2. class of 3-regular (or any m-regular) graphs

but not the class of graphs containing an isolated node.

CSPs and Non-Adaptive Query Algorithms

A constraint satisfaction problem with template H is the decision problem: Given G, is $\hom(G, H) > 0$?

CSPs and Non-Adaptive Query Algorithms

A constraint satisfaction problem with template H is the decision problem: Given G, is $\hom(G, H) > 0$?

Fact:

$\mathrm{CSP}(H) := \{G \mid \hom(G, H) > 0\}$ admits a trivial non-adaptive right algorithm ($k = 1$): Take $F_1 = H$ and $X = \{n \in \mathbb{N} \mid n > 0\}$.

CSPs and Non-Adaptive Query Algorithms

A constraint satisfaction problem with template H is the decision problem: Given G, is $\hom(G, H) > 0$?

Fact:

$\mathrm{CSP}(H) := \{G \mid \hom(G, H) > 0\}$ admits a trivial non-adaptive right algorithm ($k = 1$): Take $F_1 = H$ and $X = \{n \in \mathbb{N} \mid n > 0\}$.

Theorem (Kolaitis-W. 2022):

For every H, the class $\mathrm{CSP}(H)$ admits a k-non-adaptive left algorithm for some $k \geq 1$ iff H contains no edge.

CSPs and Non-Adaptive Query Algorithms

A constraint satisfaction problem with template H is the decision problem: Given G, is $\hom(G, H) > 0$?

Fact:

$\mathrm{CSP}(H) := \{G \mid \hom(G, H) > 0\}$ admits a trivial non-adaptive right algorithm ($k = 1$): Take $F_1 = H$ and $X = \{n \in \mathbb{N} \mid n > 0\}$.

Theorem (Kolaitis-W. 2022):

For every H, the class $\mathrm{CSP}(H)$ admits a k-non-adaptive left algorithm for some $k \geq 1$ iff H contains no edge.

- Trivial when H contains no edge: $k = 1$, take $F_1 = K_2$ (single-edge graph) and $X = \{0\}$. ($\hom(K_2, G) = 2 \times |E(G)|$.)

Proof of the Theorem

Let H contain an edge. Show that for every finite class $\mathcal{F}$ of graphs, there are $G_0 \in \mathrm{CSP}(H), G_1 \notin \mathrm{CSP}(H)$ with $\hom(F, G_0) = \hom(F, G_1)$ for $F \in \mathcal{F}$.

Proof of the Theorem

Let H contain an edge. Show that for every finite class $\mathcal{F}$ of graphs, there are $G_0 \in \mathrm{CSP}(H), G_1 \notin \mathrm{CSP}(H)$ with $\hom(F, G_0) = \hom(F, G_1)$ for $F \in \mathcal{F}$.

$\chi(H)$: chrom. num. of H,

Case 1: $\chi(H) = 2$. Then $\mathrm{CSP}(H) = \{G \mid G \text{ is 2-colorable}\}$.

Proof of the Theorem

Let H contain an edge. Show that for every finite class $\mathcal{F}$ of graphs, there are $G_0 \in \mathrm{CSP}(H), G_1 \notin \mathrm{CSP}(H)$ with $\hom(F, G_0) = \hom(F, G_1)$ for $F \in \mathcal{F}$.

$\chi(H)$: chrom. num. of H,

Case 1: $\chi(H) = 2$. Then $\mathrm{CSP}(H) = \{G \mid G \text{ is 2-colorable}\}$. Suffices to consider $\mathcal{F}$ whose graphs are connected.

Proof of the Theorem

Let H contain an edge. Show that for every finite class $\mathcal{F}$ of graphs, there are $G_0 \in \mathrm{CSP}(H), G_1 \notin \mathrm{CSP}(H)$ with $\hom(F, G_0) = \hom(F, G_1)$ for $F \in \mathcal{F}$.

$\chi(H)$: chrom. num. of H, C_n: cycle of size n, $\oplus$: disjoint union.

Case 1: $\chi(H) = 2$. Then $\mathrm{CSP}(H) = \{G \mid G \text{ is 2-colorable}\}$. Suffices to consider $\mathcal{F}$ whose graphs are connected. Choose large enough odd $n > 0$ and take $G_0 = C_{2n}, G_1 = C_n \oplus C_n$.

Proof of the Theorem

Let H contain an edge. Show that for every finite class $\mathcal{F}$ of graphs, there are $G_0 \in \mathrm{CSP}(H), G_1 \notin \mathrm{CSP}(H)$ with $\hom(F, G_0) = \hom(F, G_1)$ for $F \in \mathcal{F}$.

$\chi(H)$: chrom. num. of H, C_n: cycle of size n, $\oplus$: disjoint union.

Case 1: $\chi(H) = 2$. Then $\mathrm{CSP}(H) = \{G \mid G \text{ is 2-colorable}\}$. Suffices to consider $\mathcal{F}$ whose graphs are connected. Choose large enough odd $n > 0$ and take $G_0 = C_{2n}, G_1 = C_n \oplus C_n$.

Case 2: $\chi(H) \geq 3$. Let $n >$ the tree-width of every $F \in \mathcal{F}$.

Proof of the Theorem

Let H contain an edge. Show that for every finite class $\mathcal{F}$ of graphs, there are $G_0 \in \mathrm{CSP}(H), G_1 \notin \mathrm{CSP}(H)$ with $\hom(F, G_0) = \hom(F, G_1)$ for $F \in \mathcal{F}$.

$\chi(H)$: chrom. num. of H, C_n: cycle of size n, $\oplus$: disjoint union.

Case 1: $\chi(H) = 2$. Then $\mathrm{CSP}(H) = \{G \mid G \text{ is 2-colorable}\}$. Suffices to consider $\mathcal{F}$ whose graphs are connected. Choose large enough odd $n > 0$ and take $G_0 = C_{2n}, G_1 = C_n \oplus C_n$.

C^n: first-order counting logic with at most n variables.

Case 2: $\chi(H) \geq 3$. Let $n >$ the tree-width of every $F \in \mathcal{F}$. Then there are $G_0 \in \mathrm{CSP}(H), G_1 \notin \mathrm{CSP}(H)$ such that, successively:

G_0, G_1 satisfy same C^n-sentences (Atserias-Kolaitis-W. 2021),

Proof of the Theorem

Let H contain an edge. Show that for every finite class $\mathcal{F}$ of graphs, there are $G_0 \in \mathrm{CSP}(H), G_1 \notin \mathrm{CSP}(H)$ with $\hom(F, G_0) = \hom(F, G_1)$ for $F \in \mathcal{F}$.

$\chi(H)$: chrom. num. of H, C_n: cycle of size n, $\oplus$: disjoint union.

Case 1: $\chi(H) = 2$. Then $\mathrm{CSP}(H) = \{G \mid G \text{ is 2-colorable}\}$. Suffices to consider $\mathcal{F}$ whose graphs are connected. Choose large enough odd $n > 0$ and take $G_0 = C_{2n}, G_1 = C_n \oplus C_n$.

C^n: first-order counting logic with at most n variables.

Case 2: $\chi(H) \geq 3$. Let $n >$ the tree-width of every $F \in \mathcal{F}$. Then there are $G_0 \in \mathrm{CSP}(H), G_1 \notin \mathrm{CSP}(H)$ such that, successively:

	G_0, G_1 satisfy same C^n-sentences	(Atserias-Kolaitis-W. 2021),
$\Leftrightarrow$	$\hom(F, G_0) = \hom(F, G_1)$ for F of tree-width $< n$	(Dvořák 2010),

Proof of the Theorem

Let H contain an edge. Show that for every finite class $\mathcal{F}$ of graphs, there are $G_0 \in \mathrm{CSP}(H), G_1 \notin \mathrm{CSP}(H)$ with $\hom(F, G_0) = \hom(F, G_1)$ for $F \in \mathcal{F}$.

$\chi(H)$: chrom. num. of H, C_n: cycle of size n, $\oplus$: disjoint union.

Case 1: $\chi(H) = 2$. Then $\mathrm{CSP}(H) = \{G \mid G \text{ is 2-colorable}\}$. Suffices to consider $\mathcal{F}$ whose graphs are connected. Choose large enough odd $n > 0$ and take $G_0 = C_{2n}, G_1 = C_n \oplus C_n$.

C^n: first-order counting logic with at most n variables.

Case 2: $\chi(H) \geq 3$. Let $n >$ the tree-width of every $F \in \mathcal{F}$. Then there are $G_0 \in \mathrm{CSP}(H), G_1 \notin \mathrm{CSP}(H)$ such that, successively:

G_0, G_1 satisfy same C^n-sentences (Atserias-Kolaitis-W. 2021),
$\Leftrightarrow$ $\hom(F, G_0) = \hom(F, G_1)$ for F of tree-width $< n$ (Dvořák 2010),
$\Rightarrow$ $\hom(F, G_0) = \hom(F, G_1)$ for $F \in \mathcal{F}$.

Isomorphism and Adaptive Query Algorithms

$\cong$: isomorphic

Theorem (Chen-Flum-Liu-Xun 2022):

For $n > 0$, two graphs $F_1 = F_1(n), F_2 = F_2(n)$ can be constructed such that for all G, H of size n:

$$G \cong H \quad \text{iff} \quad \hom(F, G) = \hom(F, H) \text{ for } F \in \{F_1, F_2\}.$$

Isomorphism and Adaptive Query Algorithms

$\cong$: isomorphic

Theorem (Chen-Flum-Liu-Xun 2022):

For $n > 0$, two graphs $F_1 = F_1(n), F_2 = F_2(n)$ can be constructed such that for all G, H of size n:

$$G \cong H \quad \text{iff} \quad \hom(F, G) = \hom(F, H) \text{ for } F \in \{F_1, F_2\}.$$

Proof based on theorem by Lovász (1967):

$$G \cong H \quad \text{iff} \quad \hom(F, G) = \hom(F, H) \text{ for all } F.$$

- Sufficient to consider all F of size $\leq \min\{|V(G)|, |V(H)|\}$.

Isomorphism and Adaptive Query Algorithms

$\cong$: isomorphic

Theorem (Chen-Flum-Liu-Xun 2022):

For $n > 0$, two graphs $F_1 = F_1(n), F_2 = F_2(n)$ can be constructed such that for all G, H of size n:

$$G \cong H \quad \text{iff} \quad \hom(F, G) = \hom(F, H) \text{ for } F \in \{F_1, F_2\}.$$

Proof based on theorem by Lovász (1967):

$$G \cong H \quad \text{iff} \quad \hom(F, G) = \hom(F, H) \text{ for all } F.$$

- Sufficient to consider all F of size $\leq \min\{|V(G)|, |V(H)|\}$.

Corollary:

Three adaptive left queries $\hom(I_1, \cdot), \hom(F_1, \cdot), \hom(F_2, \cdot)$ suffice to determine, for all G and H, whether $G \cong H$.

Isomorphism and Adaptive Query Algorithms

$\cong$: isomorphic

Theorem (Chen-Flum-Liu-Xun 2022):

For $n > 0$, two graphs $F_1 = F_1(n), F_2 = F_2(n)$ can be constructed such that for all G, H of size n:

$$G \cong H \quad \text{iff} \quad \hom(F, G) = \hom(F, H) \text{ for } F \in \{F_1, F_2\}.$$

Proof based on theorem by Lovász (1967):

$$G \cong H \quad \text{iff} \quad \hom(F, G) = \hom(F, H) \text{ for all } F.$$

- Sufficient to consider all F of size $\leq \min\{|V(G)|, |V(H)|\}$.

Corollary:

Three adaptive left queries $\hom(I_1, \cdot), \hom(F_1, \cdot), \hom(F_2, \cdot)$ suffice to determine, for all G and H, whether $G \cong H$.

- $\hom(I_1, G) = |V(G)|$ (I_1: single-node graph)

Isomorphism and Adaptive Query Algorithms

$\cong$: isomorphic

Theorem (Chen-Flum-Liu-Xun 2022):

For $n > 0$, two graphs $F_1 = F_1(n), F_2 = F_2(n)$ can be constructed such that for all G, H of size n:

$$G \cong H \quad \text{iff} \quad \mathrm{hom}(F, G) = \mathrm{hom}(F, H) \text{ for } F \in \{F_1, F_2\}.$$

Proof based on theorem by Lovász (1967):

$$G \cong H \quad \text{iff} \quad \mathrm{hom}(F, G) = \mathrm{hom}(F, H) \text{ for all } F.$$

- Sufficient to consider all F of size $\leq \min\{|V(G)|, |V(H)|\}$.

Corollary:

Three adaptive left queries $\mathrm{hom}(I_1, \cdot), \mathrm{hom}(F_1, \cdot), \mathrm{hom}(F_2, \cdot)$ suffice to determine, for all G and H, whether $G \cong H$.

- $\mathrm{hom}(I_1, G) = |V(G)|$ (I_1: single-node graph)
- Optimal in number of queries made when only left homomorphism counts are allowed.

Isomorphism and Adaptive Query Algorithms

Chen et al. (2022) showed that the class of graphs containing a triangle does not admit any k-adaptive right algorithm ($k \geq 1$).

Isomorphism and Adaptive Query Algorithms

Chen et al. (2022) showed that the class of graphs containing a triangle does not admit any k-adaptive right algorithm ($k \geq 1$).

- Proof implies that no fixed number of adaptive right queries suffice to determine, for all G and H, whether $G \cong H$.

Isomorphism and Adaptive Query Algorithms

Chen et al. (2022) showed that the class of graphs containing a triangle does not admit any k-adaptive right algorithm ($k \geq 1$).

- Proof implies that no fixed number of adaptive right queries suffice to determine, for all G and H, whether $G \cong H$.

Theorem (Kolaitis-W. 2022):

For $n > 0$, a single graph $F_0 = F_0(n)$ can be constructed such that for G, H of size n:

$$G \cong H \quad \text{iff} \quad \hom(G, F_0) = \hom(H, F_0).$$

Isomorphism and Adaptive Query Algorithms

Chen et al. (2022) showed that the class of graphs containing a triangle does not admit any k-adaptive right algorithm ($k \geq 1$).

- Proof implies that no fixed number of adaptive right queries suffice to determine, for all G and H, whether $G \cong H$.

Theorem (Kolaitis-W. 2022):

For $n > 0$, a single graph $F_0 = F_0(n)$ can be constructed such that for G, H of size n:

$$G \cong H \quad \text{iff} \quad \text{hom}(G, F_0) = \text{hom}(H, F_0).$$

Corollary:

One left $\text{hom}(I_1, \cdot)$ and one adaptive right query $\text{hom}(\cdot, F_0)$ suffice to determine, for all G and H, whether $G \cong H$.

Isomorphism and Adaptive Query Algorithms

Chen et al. (2022) showed that the class of graphs containing a triangle does not admit any k-adaptive right algorithm ($k \geq 1$).

- Proof implies that no fixed number of adaptive right queries suffice to determine, for all G and H, whether $G \cong H$.

Theorem (Kolaitis-W. 2022):

For $n > 0$, a single graph $F_0 = F_0(n)$ can be constructed such that for G, H of size n:

$$G \cong H \quad \text{iff} \quad \hom(G, F_0) = \hom(H, F_0).$$

Corollary:

One left $\hom(I_1, \cdot)$ and one adaptive right query $\hom(\cdot, F_0)$ suffice to determine, for all G and H, whether $G \cong H$.

- Optimal in number of queries made when both left and right homomorphism counts are allowed.

Proof of the Theorem

Proof based on theorem by Chaudhuri and Vardi (1993):

$$G \cong H \qquad \text{iff} \qquad \hom(G, F) = \hom(H, F) \text{ for all } F.$$

- Sufficient to consider all F of size $\leq \min\{|V(G)|, |V(H)|\}$.

Proof of the Theorem

Proof based on theorem by Chaudhuri and Vardi (1993):

$$G \cong H \qquad \text{iff} \qquad \text{hom}(G, F) = \text{hom}(H, F) \text{ for all } F.$$

- Sufficient to consider all F of size $\leq \min\{|V(G)|, |V(H)|\}$.

Let $A_1, \ldots, A_s$ enumerate all graphs of size $\leq n$.

Proof of the Theorem

Proof based on theorem by Chaudhuri and Vardi (1993):

$$G \cong H \quad \text{iff} \quad \hom(G, F) = \hom(H, F) \text{ for all } F.$$

- Sufficient to consider all F of size $\leq \min\{|V(G)|, |V(H)|\}$.

Let $A_1, \ldots, A_s$ enumerate all graphs of size $\leq n$.

Goal:

Given $n > 0$, construct a graph $F_0(n)$ such that for every G of size n, $\hom(G, F_0(n))$ gives information of all $\hom(G, A_j)$.

Proof of the Theorem

Proof based on theorem by Chaudhuri and Vardi (1993):

$$G \cong H \quad \text{iff} \quad \hom(G, F) = \hom(H, F) \text{ for all } F.$$

- Sufficient to consider all F of size $\leq \min\{|V(G)|, |V(H)|\}$.

Let $A_1, \ldots, A_s$ enumerate all graphs of size $\leq n$.

Goal:

Given $n > 0$, construct a graph $F_0(n)$ such that for every G of size n, $\hom(G, F_0(n))$ gives information of all $\hom(G, A_j)$.

Observation:

Given (large) $D > 0$, every sequence $(a_0, \ldots, a_{k-1})$ of fixed length k with $0 \leq a_0, \ldots, a_{k-1} < D$ is encoded by the unique integer $a_0 \times D^0 + \cdots + a_{k-1} \times D^{k-1}$.

Proof of the Theorem

Proof based on theorem by Chaudhuri and Vardi (1993):

$$G \cong H \quad \text{iff} \quad \hom(G, F) = \hom(H, F) \text{ for all } F.$$

- Sufficient to consider all F of size $\leq \min\{|V(G)|, |V(H)|\}$.

Let $A_1, \ldots, A_s$ enumerate all graphs of size $\leq n$.

Goal:

Given $n > 0$, construct a graph $F_0(n)$ such that for every G of size n, $\hom(G, F_0(n))$ gives information of all $\hom(G, A_j)$.

Observation:

Given (large) $D > 0$, every sequence $(a_0, \ldots, a_{k-1})$ of fixed length k with $0 \leq a_0, \ldots, a_{k-1} < D$ is encoded by the unique integer $a_0 \times D^0 + \cdots + a_{k-1} \times D^{k-1}$.

- Make D-ary representation of $\hom(G, F_0(n))$ contain all $\hom(G, A_j)$ as certain digits.

Proof of the Theorem

Take $F_0(n) := \bigoplus_{j=1}^{s} (D^{e_j} \text{ disjoint copies of } A_j)$ for suitable positive integers $e_1, \ldots, e_s, D$. ($\oplus$: disjoint union)

Proof of the Theorem

Take $F_0(n) := \bigoplus_{j=1}^{s} (D^{e_j} \text{ disjoint copies of } A_j)$ for suitable positive integers $e_1, \ldots, e_s, D$. ($\oplus$: disjoint union)

By additivity and multiplicativity of $\mathrm{hom}(\cdot, \cdot)$, for arbitrary graph G of size n with connected components $G_1, \ldots, G_r$, we have

Proof of the Theorem

Take $F_0(n) := \bigoplus_{j=1}^{s} (D^{e_j} \text{ disjoint copies of } A_j)$ for suitable positive integers $e_1, \ldots, e_s, D$. ($\oplus$: disjoint union)

By additivity and multiplicativity of $\hom(\cdot,\cdot)$, for arbitrary graph G of size n with connected components $G_1, \ldots, G_r$, we have

$$\hom(G, F_0) = \sum_{e \in \mathcal{E}^+_{\leq n}} \left(\sum_{\substack{1 \leq j_1, \ldots, j_r \leq s \\ e_{j_1} + \cdots + e_{j_r} = e}} \prod_{k=1}^{r} \hom(G_k, A_{j_k}) \right) \times D^e.$$

- $\mathcal{E}^+_{\leq n}$: integers that are sums of at most n (not necessarily distinct) integers from $\{e_1, \ldots, e_s\}$.

Proof of the Theorem

Take $F_0(n) := \bigoplus_{j=1}^{s} (D^{e_j}$ disjoint copies of $A_j)$ for suitable positive integers $e_1, \ldots, e_s, D$. ($\oplus$: disjoint union)

By additivity and multiplicativity of $\hom(\cdot,\cdot)$, for arbitrary graph G of size n with connected components $G_1, \ldots, G_r$, we have

$$\hom(G, F_0) = \sum_{e \in \mathcal{E}^+_{\leq n}} \left(\sum_{\substack{1 \leq j_1, \ldots, j_r \leq s \\ e_{j_1} + \cdots + e_{j_r} = e}} \prod_{k=1}^{r} \hom(G_k, A_{j_k}) \right) \times D^e.$$

- $\mathcal{E}^+_{\leq n}$: integers that are sums of at most n (not necessarily distinct) integers from $\{e_1, \ldots, e_s\}$.

Desiderata:

1. R.H.S. of above identity is D-ary representation of $\hom(G, F_0)$

Proof of the Theorem

Take $F_0(n) := \bigoplus_{j=1}^{s} (D^{e_j} \text{ disjoint copies of } A_j)$ for suitable positive integers $e_1, \ldots, e_s, D$. ($\oplus$: disjoint union)

By additivity and multiplicativity of $\mathrm{hom}(\cdot,\cdot)$, for arbitrary graph G of size n with connected components $G_1, \ldots, G_r$, we have

$$\mathrm{hom}(G, F_0) = \sum_{e \in \mathcal{E}^+_{\leq n}} \left(\sum_{\substack{1 \leq j_1, \ldots, j_r \leq s \\ e_{j_1} + \cdots + e_{j_r} = e}} \prod_{k=1}^{r} \mathrm{hom}(G_k, A_{j_k}) \right) \times D^e.$$

- $\mathcal{E}^+_{\leq n}$: integers that are sums of at most n (not necessarily distinct) integers from $\{e_1, \ldots, e_s\}$.

Desiderata:

1. R.H.S. of above identity is D-ary representation of $\mathrm{hom}(G, F_0)$
2. D^{re_j}-digit of D-ary representation of $\mathrm{hom}(G, F_0)$ is $\mathrm{hom}(G, A_j)$ for all $1 \leq j \leq s$.

Future Directions

Investigate the expressive power of query algorithms in the variant settings below:

Future Directions

Investigate the expressive power of query algorithms in the variant settings below:

1. Undirected graphs replaced by **directed graphs** (or **relational structures** in general)

Future Directions

Investigate the expressive power of query algorithms in the variant settings below:

1. Undirected graphs replaced by **directed graphs** (or **relational structures** in general)
2. Homomorphism counts $\hom(G, H)$ replaced by their sign $\mathrm{sgn}(\hom(G, H))$

Future Directions

Investigate the expressive power of query algorithms in the variant settings below:

1. Undirected graphs replaced by **directed graphs** (or **relational structures** in general)
2. Homomorphism counts $\hom(G, H)$ replaced by their sign $\mathrm{sgn}(\hom(G, H))$
3. Allowing the number of queries to depend on input graph G