Generating Functions: An Hard Case

N. de Rugy-Altherre

Univ Paris Diderot, Sorbonne Paris CitÃ©, Institut de MathÃ©matiques de Jussieu, UMR 7586 CNRS,
F-75205 Paris, France

November 7, 2012

Generating functions

Definition

Let G be a graph and $\mathcal{P}$ a graph property (a class of graphs closed under isomorphism). The *generating function* corresponding to G and $\mathcal{P}$ is the following polynomial:

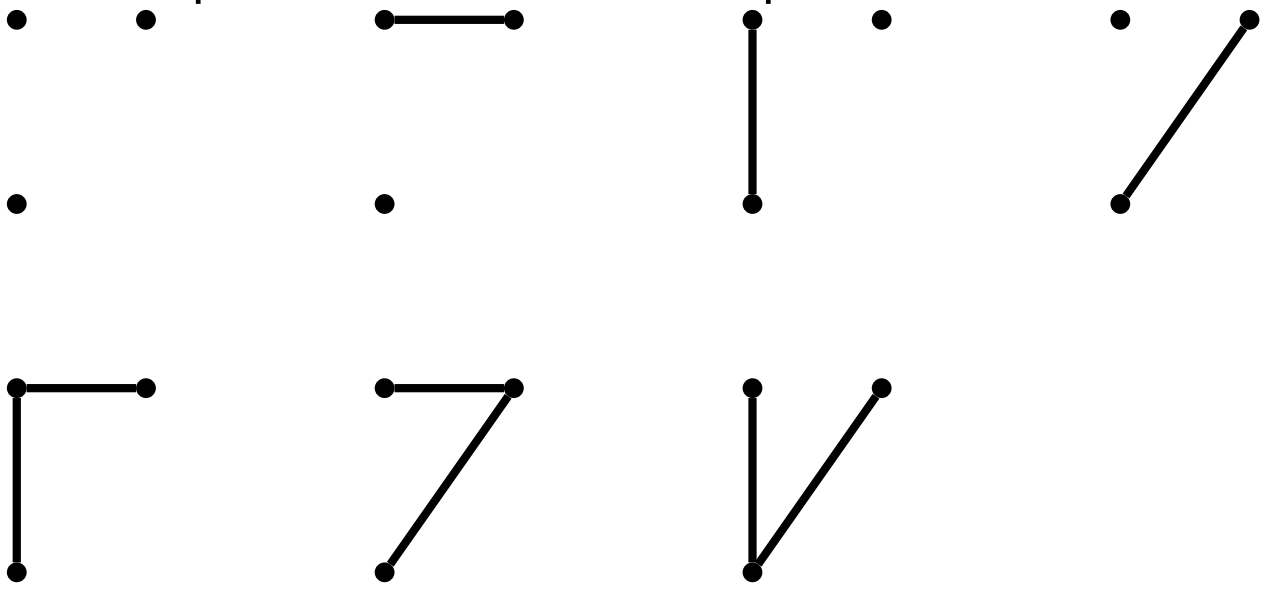
$$GF(G, \mathcal{P})(\bar{x}) = \sum_{E'} \prod_{e \in E'} x_e$$

Where the sum is over every $E' \subseteq E$ such that $(V(G), E')$ has the property $\mathcal{P}$.

Generating functions

For example, for a graph H, let Hom_H be the property of being homomorphic to H.

Graphs of size 3 homomorphic to $H =$  .



Generating functions

For example, for a graph H, let Hom_H be the property of being homomorphic to H.

Graphs of size 3 homomorphic to $H =$.

$GF(K_3, Hom_{\bullet\!-\!\bullet}) = 1 + x_{1,2} + x_{1,3} + x_{2,3} + x_{1,2}x_{1,3} + x_{1,2}x_{2,3} + x_{1,3}x_{2,3}$

Generating functions: link between different notions of complexity

- **Counting**
  - $GF(G, \mathcal{P})(\bar{1}) = |\{G' \text{ spanning subgraph of } G \text{ of size } n \text{ in } \mathcal{P}\}|$

Generating functions: link between different notions of complexity

- **Counting**
  - $GF(G, \mathcal{P})(\bar{1}) = |\{G' \text{ spanning subgraph of } G \text{ of size } n \text{ in } \mathcal{P}\}|$
  - Let G' be a spanning subgraph of G and $\bar{x}^{G'}$ be the variables of G where those which represent a edge in G and not in G' is evaluated to 1. Then $GF(G, \mathcal{P})(\bar{x}^{G'}) = GF(G', \mathcal{P})$.

Generating functions: link between different notions of complexity

- **Counting**
  - $GF(G, \mathcal{P})(\bar{1}) = |\{G' \text{ spanning subgraph of } G \text{ of size } n \text{ in } \mathcal{P}\}|$
  - Let G' be a spanning subgraph of G and $\bar{x}^{G'}$ be the variables of G where those which represent a edge in G and not in G' is evaluated to 1. Then $GF(G, \mathcal{P})(\bar{x}^{G'}) = GF(G', \mathcal{P})$.
- **Enumeration**

Generating functions: link between different notions of complexity

- **Counting**
  - $GF(G, \mathcal{P})(\bar{1}) = |\{G' \text{ spanning subgraph of } G \text{ of size } n \text{ in } \mathcal{P}\}|$
  - Let G' be a spanning subgraph of G and $\bar{x}^{G'}$ be the variables of G where those which represent a edge in G and not in G' is evaluated to 1. Then $GF(G, \mathcal{P})(\bar{x}^{G'}) = GF(G', \mathcal{P})$.
- **Enumeration**
  - Interpolation

Generating functions: link between different notions of complexity

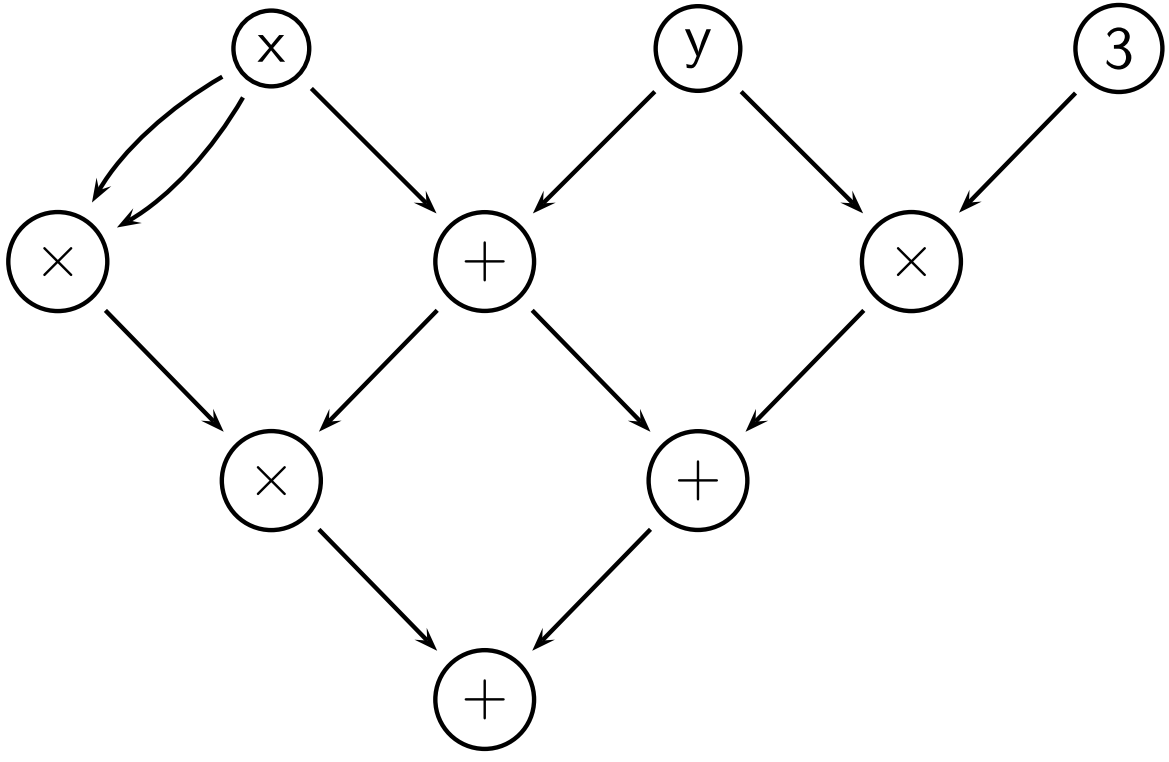
- **Counting**
  - $GF(G, \mathcal{P})(\bar{1}) = |\{G' \text{ spanning subgraph of } G \text{ of size } n \text{ in } \mathcal{P}\}|$
  - Let G' be a spanning subgraph of G and $\bar{x}^{G'}$ be the variables of G where those which represent a edge in G and not in G' is evaluated to 1. Then $GF(G, \mathcal{P})(\bar{x}^{G'}) = GF(G', \mathcal{P})$.
- **Enumeration**
  - Interpolation
  - If $GF(G, \mathcal{P})$ is computable in polynomial time, then the subgraphs of G in $\mathcal{P}$ can be enumerated in randomized polynomial delay.

Algebraic Complexity

Definition

An *arithmetic circuit* is an acyclic directed graph with indegree at maximum 2. The nodes with indegree 0 are labeled with constants of a field $\mathbb{K}$ or variables; the one with indegree 2 are labeled with $+$ and $\times$.

Algebraic Complexity



$$f(x, y) = x^3 + x^2 y + 3xy + 3y^2$$

Algebraic Complexity

Definition

- The *size* of an arithmetic circuit is the number of operational gates.

Algebraic Complexity

Definition

- The *size* of an arithmetic circuit is the number of operational gates.
- A p-family (f_n) is a sequence of polynomials such that the number of variables as well as the degree of f_n are polynomially bounded in n.

Algebraic Complexity

Definition

- The *size* of an arithmetic circuit is the number of operational gates.
- A p-family (f_n) is a sequence of polynomials such that the number of variables as well as the degree of f_n are polynomially bounded in n.
- A p-family (f_n) of polynomials is in VP if it can be computed by a family of arithmetic circuits of polynomial size in n.

Algebraic Complexity

Definition

- The *size* of an arithmetic circuit is the number of operational gates.
- A p-family (f_n) is a sequence of polynomials such that the number of variables as well as the degree of f_n are polynomially bounded in n.
- A p-family (f_n) of polynomials is in VP if it can be computed by a family of arithmetic circuits of polynomial size in n.
- A p-family (f_n) is in VNP if there is a family (g_n) in VP such that

$$f_n(\bar{x}) = \sum_{\bar{\epsilon} \in \{0,1\}^n} g_n(\bar{\epsilon}, \bar{x})$$

Generating functions easily computable

- $GF(K_n, \mathcal{P}_{triv})$ is very easy (in VAC_0).

Generating functions easily computable

- $GF(K_n, \mathcal{P}_{triv})$ is very easy (in VAC_0).
- $GF(K_n, \{\text{trees}\})$ is easier than the determinant for any sequence of graph G_n (i.e., in VP_{ws}).

Generating functions easily computable

- $GF(K_n, \mathcal{P}_{triv})$ is very easy (in VAC_0).
- $GF(K_n, \{\text{trees}\})$ is easier than the determinant for any sequence of graph G_n (i.e., in VP_{ws}).
- $\mathcal{DI}$ is the graph property expressing that any connected component has exactly two vertices (a perfect matching). Then for any sequence of planar graphs G_n, $GF(G_n, \mathcal{DI})$ is computable in polynomial time (i.e. in VP).

Generating functions easily computable

- $GF(K_n, \mathcal{P}_{triv})$ is very easy (in VAC_0).
- $GF(K_n, \{\text{trees}\})$ is easier than the determinant for any sequence of graph G_n (i.e., in VP_{ws}).
- $\mathcal{DI}$ is the graph property expressing that any connected component has exactly two vertices (a perfect matching). Then for any sequence of planar graphs G_n, $GF(G_n, \mathcal{DI})$ is computable in polynomial time (i.e. in VP).
- If $\mathcal{P}$ is MS_2-definable and G_n of bounded tree-width, then $GF(G_n, \mathcal{P})$ is computable in polynomial time (i.e., in VP).

Generating functions hardly computable

- $\mathcal{DI}$ is the graph property expressing that any connected component has exactly two vertices.. Then $\mathrm{PER}(\bar{x}) = GF(K_n, \mathcal{DI})$ is VNP-complete.

Generating functions hardly computable

- $\mathcal{DI}$ is the graph property expressing that any connected component has exactly two vertices.. Then $\mathrm{PER}(\bar{x}) = GF(K_n, \mathcal{DI})$ is VNP-complete.
- $\mathcal{MD}$ is the graph property expressing that any connected component has at most two vertices.. Then $\mathrm{PER}^*(\bar{x}) = GF(K_n, \mathcal{MD})$ is VNP-complete.

Generating functions hardly computable

- $\mathcal{DI}$ is the graph property expressing that any connected component has exactly two vertices.. Then $\mathrm{PER}(\bar{x}) = GF(K_n, \mathcal{DI})$ is VNP-complete.
- $\mathcal{MD}$ is the graph property expressing that any connected component has at most two vertices.. Then $\mathrm{PER}^*(\bar{x}) = GF(K_n, \mathcal{MD})$ is VNP-complete.
- $\mathcal{CL}$ is the graph property expressing that a connected component is a clique and the others have only one vertex. Then $GF(K_n, \mathcal{CL})$ is VNP-complete.

Generating functions hardly computable

- $\mathcal{DI}$ is the graph property expressing that any connected component has exactly two vertices.. Then $\mathrm{PER}(\bar{x}) = GF(K_n, \mathcal{DI})$ is VNP-complete.
- $\mathcal{MD}$ is the graph property expressing that any connected component has at most two vertices.. Then $\mathrm{PER}^*(\bar{x}) = GF(K_n, \mathcal{MD})$ is VNP-complete.
- $\mathcal{CL}$ is the graph property expressing that a connected component is a clique and the others have only one vertex. Then $GF(K_n, \mathcal{CL})$ is VNP-complete.
- Cycle format

Generating functions hardly computable

- $\mathcal{DI}$ is the graph property expressing that any connected component has exactly two vertices.. Then $\mathrm{PER}(\bar{x}) = GF(K_n, \mathcal{DI})$ is VNP-complete.
- $\mathcal{MD}$ is the graph property expressing that any connected component has at most two vertices.. Then $\mathrm{PER}^*(\bar{x}) = GF(K_n, \mathcal{MD})$ is VNP-complete.
- $\mathcal{CL}$ is the graph property expressing that a connected component is a clique and the others have only one vertex. Then $GF(K_n, \mathcal{CL})$ is VNP-complete.
- Cycle format
- Graph factors

Generating functions hardly computable

- $\mathcal{DI}$ is the graph property expressing that any connected component has exactly two vertices.. Then $\mathrm{PER}(\bar{x}) = GF(K_n, \mathcal{DI})$ is VNP-complete.
- $\mathcal{MD}$ is the graph property expressing that any connected component has at most two vertices.. Then $\mathrm{PER}^*(\bar{x}) = GF(K_n, \mathcal{MD})$ is VNP-complete.
- $\mathcal{CL}$ is the graph property expressing that a connected component is a clique and the others have only one vertex. Then $GF(K_n, \mathcal{CL})$ is VNP-complete.
- Cycle format
- Graph factors
- ...

A wanted general theorem for generating functions

Theorem (Imaginary)

Every generating function of a graph property expressible in MS_2 is VNP*-complete when computed on general graphs if and only if the graph property satisfy a condition C. If not, it is in VP.*

A wanted general theorem for generating functions

Theorem (Imaginary)

Every generating function of a graph property expressible in MS_2 is VNP-complete when computed on general graphs if and only if the graph property satisfy a condition C. If not, it is in VP.

Theorem (Real)

Every generating function of a graph property that can be express as an homomorphism with fixed target is VNP-complete when computed on general graphs.

Algebraic Complexity

Definition

- A p-family (g_n) is a *p-reduction* of (f_n), write $(g_n) \leq_p (f_n)$, is there is a polynomial p and $a_1, \ldots, a_{p(n)}$ variables or constants such that

$$g_n(\bar{x}) = f_{p(n)}(\bar{a})$$

Algebraic Complexity

Definition

- A p-family (g_n) is a *p-reduction* of (f_n), write $(g_n) \leq_p (f_n)$, is there is a polynomial p and $a_1, \ldots, a_{p(n)}$ variables or constants such that

$$g_n(\bar{x}) = f_{p(n)}(\bar{a})$$

- A p-family (g_n) is a *c-reduction* of (f_n), write $(g_n) \leq_c (f_n)$, is there is a polynomial p an arithmetic circuit of polynomial size with oracle $f_{p(n)}$ that compute g_n.

Algebraic Complexity

Definition

- A p-family (g_n) is a *p-reduction* of (f_n), write $(g_n) \leq_p (f_n)$, is there is a polynomial p and $a_1, \ldots, a_{p(n)}$ variables or constants such that

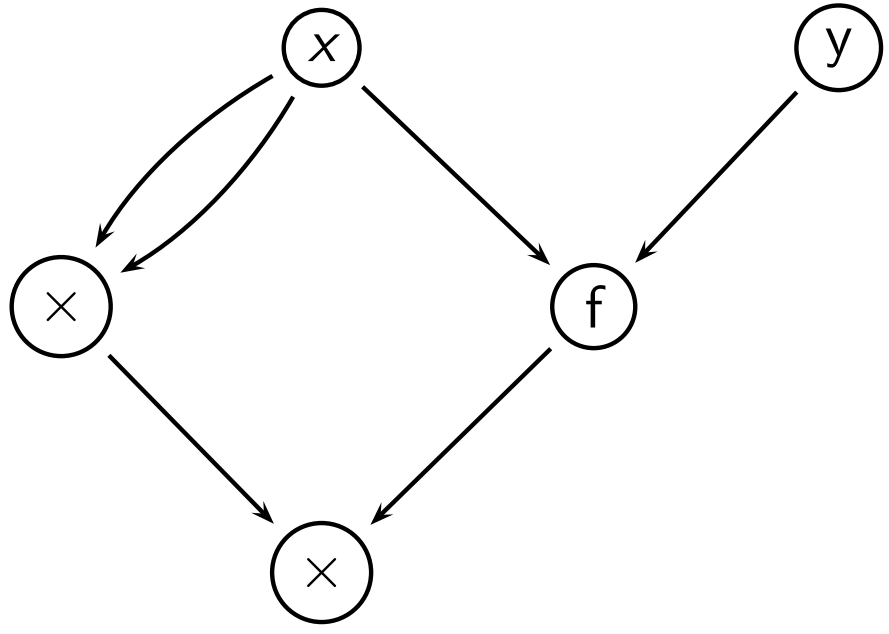
$$g_n(\bar{x}) = f_{p(n)}(\bar{a})$$

- A p-family (g_n) is a *c-reduction* of (f_n), write $(g_n) \leq_c (f_n)$, is there is a polynomial p an arithmetic circuit of polynomial size with oracle $f_{p(n)}$ that compute g_n.
- A p-family (f_n) is VNP-*complete* for c-reductions is for any family $(g_n) \in \mathrm{VNP}$,

$$(g_n) \leq_c (f_n)$$

Algebraic Complexity

$$g(x, y) = x^2(x^3 + x^2y + 3xy + 3y^2)$$



If $f(x, y) = x^3 + x^2y + 3xy + 3y^2$, then $g \leq_c f$.

Algebraic Complexity

- Let (f_n) be a p-family. Let us write $\mathbf{HC}_k(f_n)$ for the homogeneous component of degree k of f_n.
- There is some constants $w_{i,k} \in \mathbb{R}$ such that

$$\mathbf{HC}_k(f_n)(\bar{x}) = \sum_{i=0}^{d_n} w_{i,k} f^n(2^i \bar{x})$$

- Then for any sequence of integers (k_n),

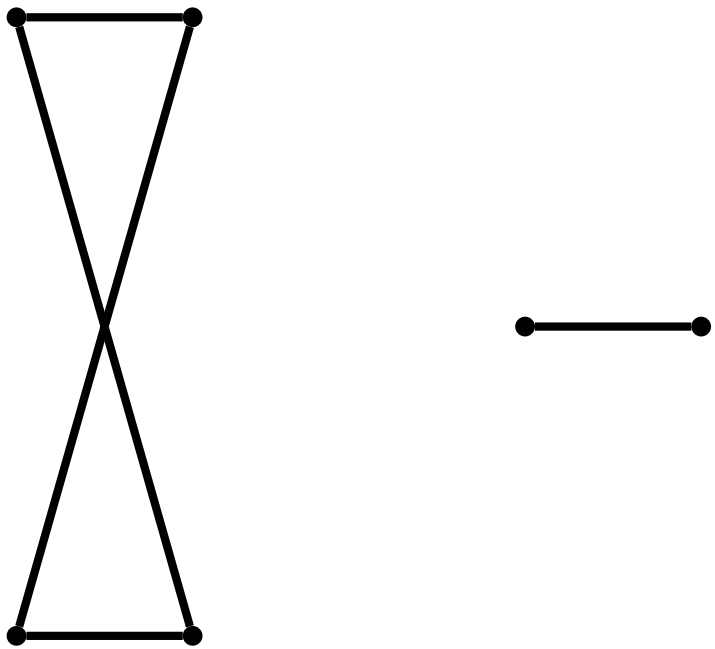
$$(\mathbf{HC}_{k_n}(f_n)) \leq_c (f_n)$$

Graph homomorphisms

Definition

Let G and H be two graphs and let ϕ an application from $V(G)$ to $V(H)$. It is an homomorphism if:

$$\forall u, v \in V(G), (u, v) \in E(G) \Rightarrow (\phi(u), \phi(v)) \in E(F)$$

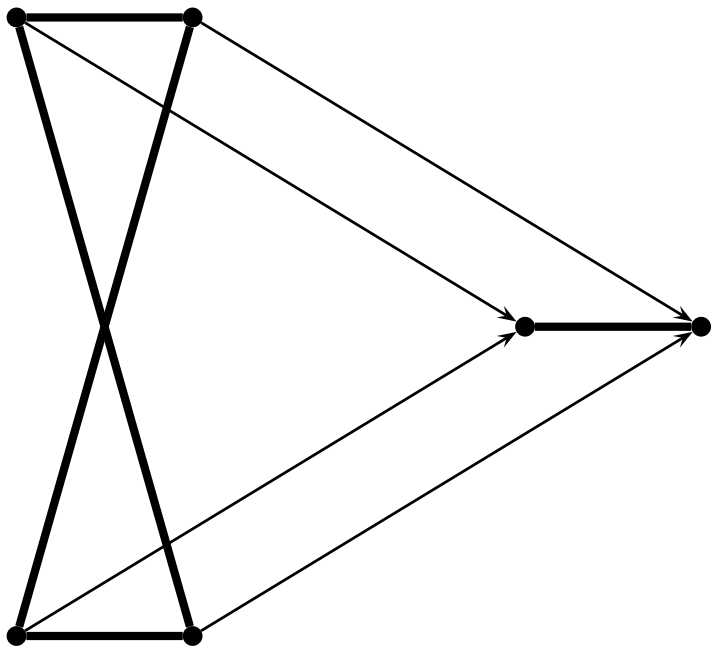


Graph homomorphisms

Definition

Let G and H be two graphs and let ϕ an application from $V(G)$ to $V(H)$. It is an homomorphism if:

$$\forall u, v \in V(G), (u, v) \in E(G) \Rightarrow (\phi(u), \phi(v)) \in E(F)$$



Homomorphism polynomials

Definition

Let H be a graph and $Hom(H)$ be the property expressing that a graph has a connected component homomorphic to H and all other reduced to a single vertex. Then we write

$$f_n^H((x_e)_{e\in E(K_n)}) := GF(K_n, Hom_H) = \sum_{E'} \prod_{e\in E'} x_e$$

Where the sum is over every $E' \subseteq E(G)$ such that the graph $(V(G), E')$ is connected and homomorphic to H.

Homomorphism polynomials

Theorem

Let H be a graph. Then, over $\mathbb{Q}$

- *If H has a loop or no edges, (f_n^H) is in VAC_0.*
- *Else (f_n^H) is VNP-complete under c-reductions.*

The first step of the proof

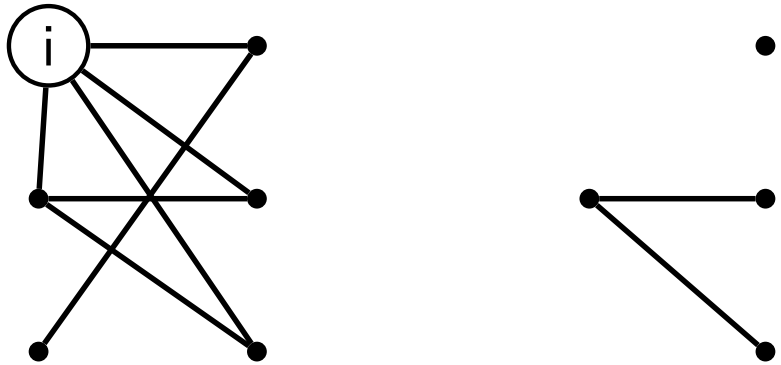
Proposition

For any graph H with no loops and at least one edge,

$$(f_n^{\bullet\!-\!\bullet}) \leq_c \left(f_n^H\right)$$

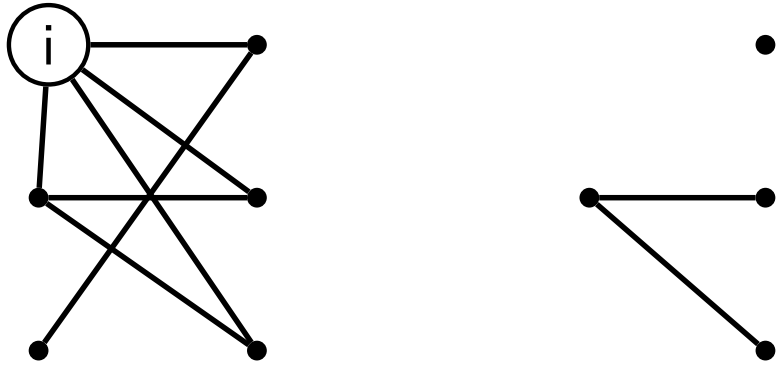
The first step of the proof

- Let H_i be the subgraph of H built by the neighbour of i and $\tilde{H} = \cup_{i \in V(H)} H_i$.

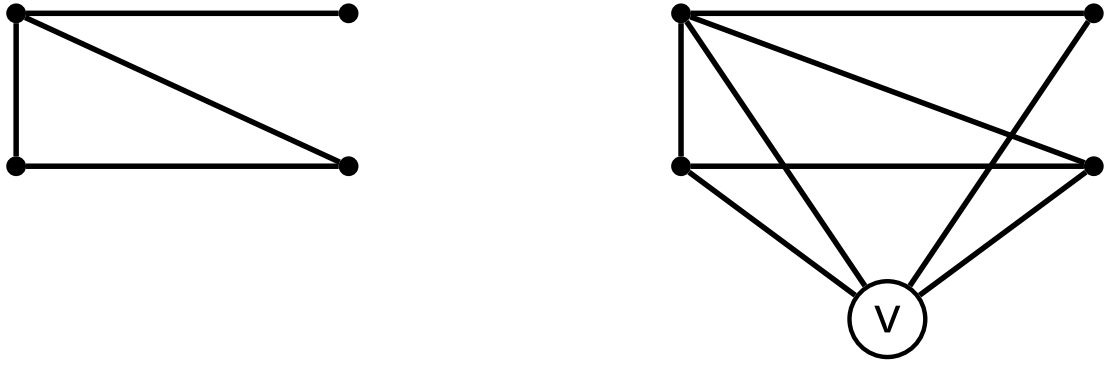


The first step of the proof

- Let H_i be the subgraph of H built by the neighbour of i and $\tilde{H} = \cup_{i \in V(H)} H_i$.

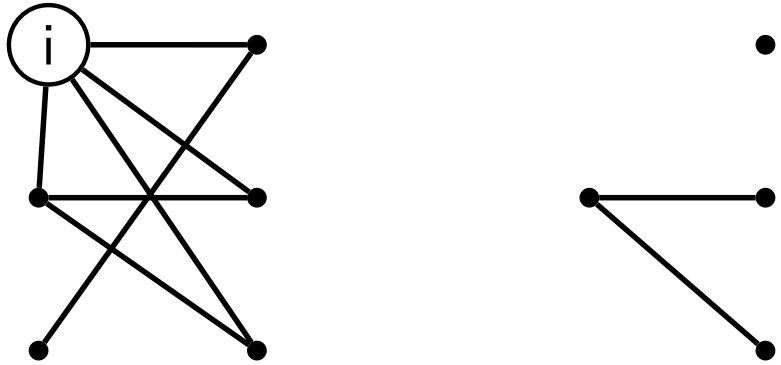


- Let us write G' the graph G with a new vertex v linked to every others.

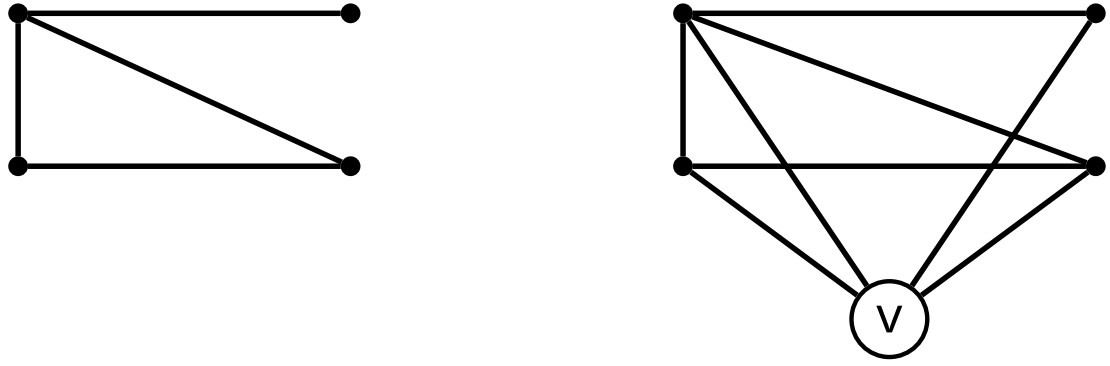


The first step of the proof

- Let H_i be the subgraph of H built by the neighbour of i and $\tilde{H} = \cup_{i \in V(H)} H_i$.



- Let us write G' the graph G with a new vertex v linked to every others.



- G' is homomorphic to H iff G is to $\tilde{H}$.

The first step of the proof

$$f^H_{n+1}(\bar{x}, x_{1,n+1}, \ldots, x_{n,n+1}) = \sum_{G \text{ homomorphic to } H} \bar{x}^G$$

$$f^H_{n+1}(\bar{x}, y, \ldots, y) = \sum_{G \text{ homomorphic to } H} \bar{x}^G y^{m(G)}$$

Where $m(G)$ is the number of edges from $n+1$ to an other vertex.

$$\begin{aligned}
\mathbf{HC}^y_n \left(f^H_{n+1}\right)(\bar{x}, 1, \ldots, 1) &= \sum_{G' \text{ homomorphic to } H} \bar{x}^G \\
&= \sum_{G \text{ homomorphic to } \tilde{H}} \bar{x}^G \\
&= f^{\tilde{H}}_n(\bar{x})
\end{aligned}$$

Therefore, $(f^{\tilde{H}}_n) \leq_c (f^H_n)$

The first step of the proof

Definition

The *maximal degree* of a graph is the maximal number of edges coming from a single vertex.

The maximal degree of $\tilde{H}$ is strictly lower than the maximal degree of H.

The first step of the proof

Definition

The *maximal degree* of a graph is the maximal number of edges coming from a single vertex.

The maximal degree of $\tilde{H}$ is strictly lower than the maximal degree of H.

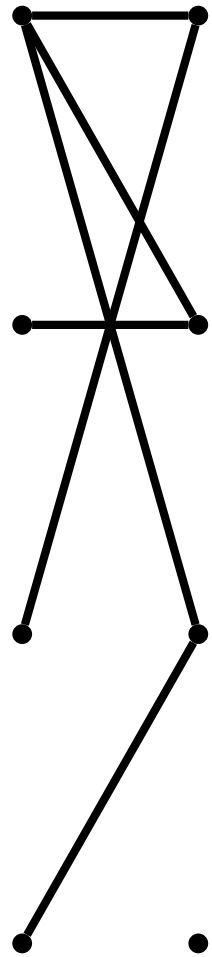
Proposition

For any graph H with no loops and at least one edge,

$$(f_n^{\bullet\!-\!\bullet}) \leq_c \left(f_n^H\right)$$

Proof that $(f_n^{\bullet\!-\!\bullet})$ is VNP-complete

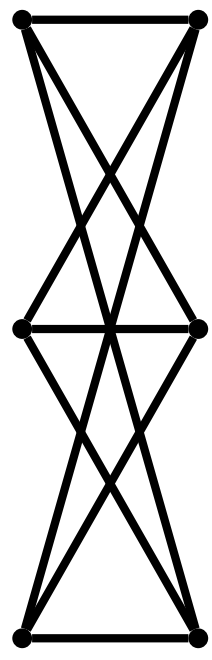
$f_n^{\bullet\!-\!\bullet}$ enumerates every bipartite graphs on n vertices, i.e., if $\mathcal{BIP}$ is the graph property of being bipartite, $f_n^{\bullet\!-\!\bullet} = GF(K_n, \mathcal{BIP})$.



Proof that $(f_n^{\bullet\!-\!\bullet})$ is VNP-complete

G_n enumerates every complete bipartite graphs on $2n$ vertices.
i.e., if $\mathcal{CBIP}$ is the graph property of being a complete bipartite graph, $G_n = GF(K_{2n}, \mathcal{CBIP})$..

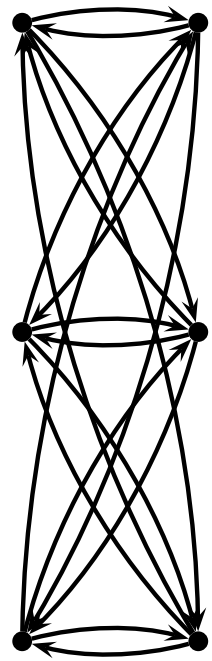
G_n



Proof that $(f_n^{\bullet\!-\!\bullet})$ is VNP-complete

Cut_n^2 enumerates every oriented complete bipartite graphs on $2n$ vertices.

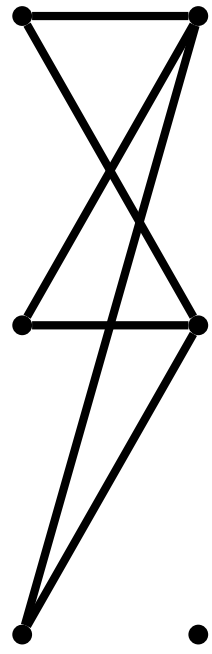
$\mathbf{Cut}_n^2$



Proof that $(f_n^{\bullet-\bullet})$ is VNP-complete

F_n enumerates every complete bipartite graphs on a subset of $2n$ vertices. i.e., if $\mathcal{BIPSCOM}$ is the graph property of having a connect component complete bipartite and every other reduced to a vertex, $F_n = GF(K_n, \mathcal{BIPSCOM})$..

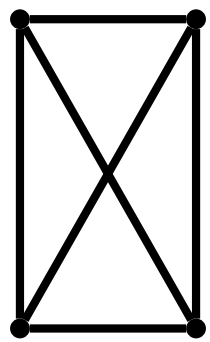
F_n



Proof that $(f_n^{\bullet\!-\!\bullet})$ is VNP-complete

GF(K_n, clique) enumerates every cliques on n vertices.

G_n



Conclusion

$$(\mathsf{GF}(K_n, \mathsf{clique})) \leq_c (F_n) \leq_p (\mathsf{Cut}^2_n) \leq_c (G_n) \leq_c (f_n^{\bullet\!-\!\bullet}) \leq_c (f_n^H)$$

Conclusion

$$(\mathsf{GF}(K_n, \mathsf{clique})) \leq_c (F_n) \leq_p (\mathsf{Cut}^2_n) \leq_c (G_n) \leq_c (f_n^{\bullet\!\!-\!\!\bullet}) \leq_c (f_n^H)$$

$$\mathsf{Cut}^q_n(\bar{x}) = \sum_{A \cup B = [n]} \prod_{i \in A, j \in B} x^q_{i,j}$$

Conclusion

$$(\mathsf{GF}(K_n, \mathsf{clique})) \leq_c (F_n) \leq_p (\mathsf{Cut}_n^2) \leq_c (G_n) \leq_c (f_n^{\bullet\!-\!\bullet}) \leq_c (f_n^H)$$

$$\mathsf{Cut}_n^q(\bar{x}) = \sum_{A \cup B = [n]} \prod_{i \in A, j \in B} x_{i,j}^q$$

Theorem (Burgisser, 5.22)

Let q be a power of the prime p. The family of cut enumerators (Cut^q) over $\mathbb{F}_q$ is neither VP nor VNP-complete with respect to c-reductions, provided $Mod_p\mathrm{NP} \not\subseteq \mathrm{P}/poly$.

Conclusion

$$(\mathsf{GF}(K_n, \mathsf{clique})) \leq_c (F_n) \leq_p (\mathsf{Cut}_n^2) \leq_c (G_n) \leq_c (f_n^{\bullet\!-\!\bullet}) \leq_c (f_n^H)$$

$$\mathsf{Cut}_n^q(\bar{x}) = \sum_{A \cup B = [n]} \prod_{i \in A, j \in B} x_{i,j}^q$$

Theorem (Burgisser, 5.22)

Let q be a power of the prime p. The family of cut enumerators (Cut^q) over $\mathbb{F}_q$ is neither VP nor VNP-complete with respect to c-reductions, provided $\mathrm{Mod}_p\mathrm{NP} \not\subseteq \mathrm{P/poly}$.

Problem (Burgisser, 5.2)

Is Cut^2, interpreted as family over the rationals, VNP-complete?

Conclusion

$$(\mathsf{GF}(K_n, \mathsf{clique})) \leq_c (F_n) \leq_p (\mathsf{Cut}^2_n) \leq_c (G_n) \leq_c (f_n^{\bullet\!-\!\bullet}) \leq_c (f_n^H)$$

$$\mathsf{Cut}^q_n(\bar{x}) = \sum_{A \cup B = [n]} \prod_{i \in A, j \in B} x^q_{i,j}$$

Theorem (Burgisser, 5.22)

Let q be a power of the prime p. The family of cut enumerators (Cut^q) over $\mathbb{F}_q$ is neither VP nor VNP-complete with respect to c-reductions, provided $Mod_p\mathrm{NP} \not\subseteq \mathrm{P/poly}$.

Problem (Burgisser, 5.2)

Is Cut^2, interpreted as family over the rationals, VNP-complete?

This answer is yes.

Conclusion

$$(\mathsf{GF}(K_n, \mathsf{clique})) \leq_c (F_n) \leq_p (\mathsf{Cut}_n^2) \leq_c (G_n) \leq_c (f_n^{\bullet\!-\!\bullet}) \leq_c (f_n^H)$$

$$\mathsf{Cut}_n^q(\bar{x}) = \sum_{A \cup B = [n]} \prod_{i \in A, j \in B} x_{i,j}^q$$

Theorem (Burgisser, 5.22)

Let q be a power of the prime p. The family of cut enumerators (Cut^q) over $\mathbb{F}_q$ is neither VP nor VNP-complete with respect to c-reductions, provided $Mod_p\mathrm{NP} \not\subseteq \mathrm{P/poly}$.

Problem (Burgisser, 5.2)

Is Cut^2, interpreted as family over the rationals, VNP-complete?

This answer is yes.
Thank you!