Notions of Width

Variables, Pebbles, Supports

Anuj Dawar

Department of Computer Science and Technology, University of Cambridge

LICS, Singapore, 23 June 2025

What is this talk about?

Start with a straightforward question:

> *What is the smallest number of variables with which you can equivalently write a given formula φ of first-order logic?*

We always assume we have formulas in a fixed *relational vocabulary* (say the vocabulary of *graphs*):

$$x = y \mid E(x, y) \mid \varphi \wedge \psi \mid \varphi \vee \psi \mid \neg\varphi \mid \exists x\varphi \mid \forall x\varphi$$

Each formula $\varphi(\mathbf{x})$ defines a *query*, i.e. a map that takes a finite graph G to a *relation* $\varphi^G \subseteq G^{\mathbf{x}}$.
Why do we care about the total number of variables?
The answer will take us on a tour touching on *combinatorics*; *conjunctive queries*; *categories* and *circuit complexity*.

Flashback: LICS 1990

Ph.G. Kolaitis and M.Y. Vardi, *0-1 Laws for Infinitary Logics*, LICS 1990, Philadelphia.

This proves a 0-1-law for $L^{\omega}_{\infty\omega}$

> *the closure of first-order logic under infinitary conjunctions and disjunctions, but limiting each formula to a finite number of distinct variables.*

Every formula of FP (*fixed-point logic*) is equivalent *over finite structures* to a formula of $L^{\omega}_{\infty\omega}$.

Variable-confined Logics

L^k— formulas of first-order logic using only the variables $x_1, \ldots, x_k$.
$L^k_{\infty\omega}$—closure of L^k under infinitary conjunctions and disjunctions.
$L^\omega_{\infty\omega} = \bigcup_k L^k_{\infty\omega}$

Why do we care about the *syntactic names* of the variables that appear in a formula?

Limiting the number of variables limits the maximum number of *free variables* that can appear in any subformula.

Conjunctive Queries

A *conjunctive query* is a first-order formula constructing using only *conjunction* and *existential quantification*.

Consider the query (in the language of *directed graphs*) saying *"there is a walk of length five"*.
In *prenex normal form* this requires *six* variables

$$\exists x_1 \cdots \exists x_6 (E(x_1, x_2) \wedge \cdots \wedge E(x_5, x_6)).$$

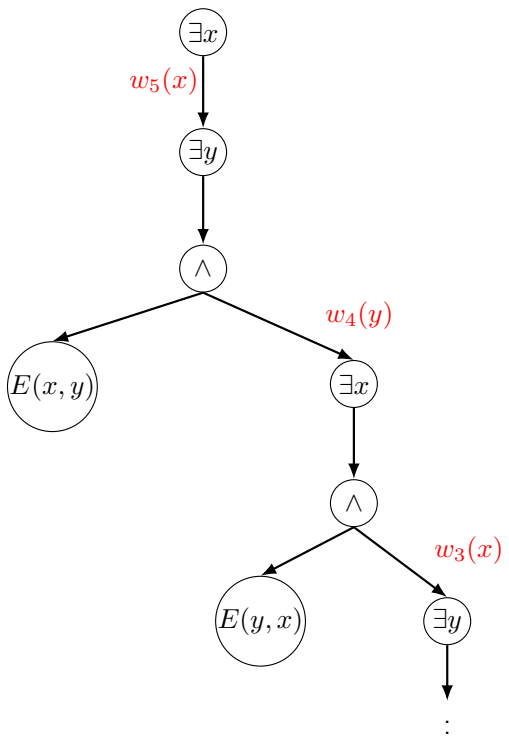
but, can be formulated with just *two*:

$$\exists x \exists y (E(x, y) \wedge \exists x (E(y, x) \wedge \exists y (\cdots))).$$

Query Plans

Formulating the query with a small number of variables allows for a *query plan* with small *intermediate relations*.

$$\exists x \exists y (E(x,y) \wedge \exists x (E(y,x) \wedge \exists y (\cdots))).$$



Chandra-Merlin

Conjunctive query evalutation is the same as *structure homomorphism*.

(Chandra Merlin 1977)

From a conjunctive query φ, we construct a structure M_φ such that for any $\mathbb{A}$

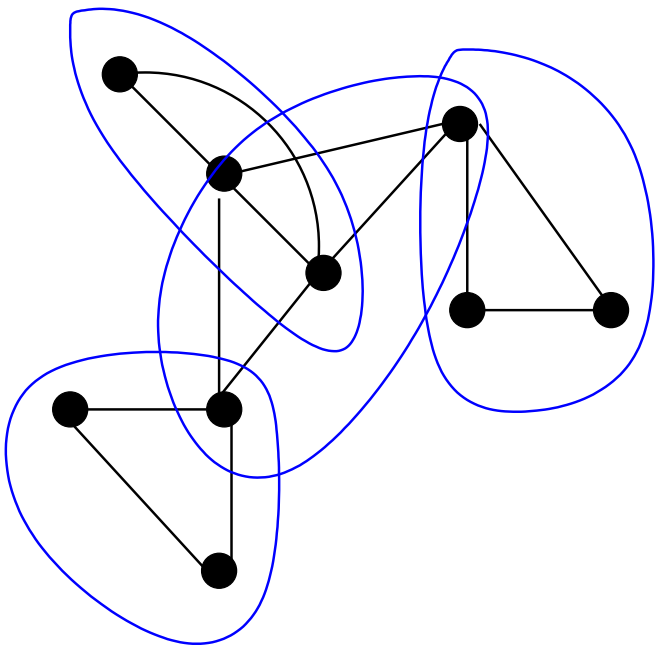
$$\mathbb{A} \models \varphi \text{ if, and only if, } M_\varphi \longrightarrow \mathbb{A}$$

Put φ in prenex normal form; strip away the existential quantifiers; view the *resulting matrix* as a structure on the set of variables.

To minimize the number of variables required to write φ it is useful to look at natural *connectivity properties* of M_φ.

Treewidth

The *treewidth* of an undirected graph is a measure of how tree-like the graph is.
A graph has treewidth k if it can be covered by subgraphs of at most $k+1$ nodes in a tree-like fashion.



Treewidth and the Number of Variables

The *treewidth* of a graph is an important combinatorial parameter much studied in *structural graph theory* and *parameterized algorithms*.

We are interested in the following property:

> *If M_φ has treewidth less than k then φ can be equivalently written with at most k variables.*
>
> **(Dalmau, Kolaitis, Vardi 2002)**

This is useful in *query optimization*.

Fixed-Point Logic

FP—*fixed-point logic* is the extension of first-order logic with a *recursion* mechanism.

A query on finite *ordered* structures is definable in FP if, and only if, it is computable in *polynomial time*. **(Immerman, Vardi 1982/1986)**

In the absence of order, FP cannot express simple *counting* properties.

This leads to the study of FPC—*fixed-point logic with counting*

Just as FP formulas can be translated into $L^{\omega}_{\infty\omega}$, so formulas of FPC translate to $C^{\omega}_{\infty\omega}$—*infinitary logic with counting*

> *The key is that recursion can be unfolded into an infinitary disjunction while keeping the number of variables bounded.*

Logics with Counting

C^k is the logic obtained from *first-order logic* by allowing:

- *counting quantifiers*: $\exists^i x\, \varphi$; and
- only the variables $x_1, \ldots . x_k$.

Every formula of C^k is equivalent to a formula of first-order logic, albeit one with more variables.

For every sentence φ of FPC, there is a k such that φ is equivalent to an *infinite disjunction* of formulas of C^k.

In particular, structures which cannot be distinguished in C^k cannot be be distinguished by φ.

Weisfeiler-Leman

For a pair of structures $\mathbb{A}$ and $\mathbb{B}$, we write $\mathbb{A} \equiv^k \mathbb{B}$ to denote that they are not distinguished by any sentence of C^k.

$\mathbb{A} \equiv^k \mathbb{B}$ is decidable in time $n^{O(k)}$.

This is an equivalence relation that has been much studied in connection with the *graph isomorphism problem*
It has many equivalent characterisations arising from
combinatorics; *logic*; *algebra*; *linear optimization*

It was proved **(Cai, Fürer, Immerman 1992)** that there is no k such that $\equiv^k$ is the same as *isomorphism*.

Homomorphism Counts

For two finite τ-structures $\mathbb{A}$ and $\mathbb{B}$, we have: $\mathbb{A} \equiv^k \mathbb{B}$ *if, and only if,* for every finite τ-structure $\mathbb{C}$ of *treewidth* less than k,

$$|\{h : \mathbb{C} \xrightarrow{\text{hom}} \mathbb{A}\}| = |\{h : \mathbb{C} \xrightarrow{\text{hom}} \mathbb{B}\}|$$

(Dvořák 2010)

If $\mathbb{A}$ and $\mathbb{B}$ are distinguished by *any* formula of C^k, then they are distinguished by one of the form $\exists^{\geq t}\mathbf{x}\theta$ where

> θ *is the matrix (in prenex normal form) of a conjunctive query that can be writtten with at most* k *variables.*

Any formula of $C^\omega_{\infty\omega}$ is an infinite *Boolean combination* of such formulas.

Pebbling Comonad

The *pebbling comonads* (defined in **(Abramsky, D, Wang 2017)** place much of this in a *categorical* framework.

For each k, we have a map that takes a structure $\mathbb{A}$ to $\mathrm{P}_k\mathbb{A}$.
We can think of $\mathrm{P}_k\mathbb{A}$ as a width k *tree unfolding* of $\mathbb{A}$.

There is a morphism from $\mathrm{P}_k\mathbb{A}$ to $\mathbb{B}$ *if, and only if,* every width k conjunctive query satisfied in $\mathbb{A}$ is satisfied in $\mathbb{B}$.

There is an isomorphism between $\mathrm{P}_k\mathbb{A}$ and $\mathrm{P}_k\mathbb{B}$ *if, and only if,* $\mathbb{A} \equiv^k \mathbb{B}$.

There is a *coalgebra* $\mathbb{A} \to \mathrm{P}_k\mathbb{A}$ *if, and only if,* the treewidth of $\mathbb{A}$ is less than k.

Combinatorial Categories

Def.
A *locally finite* category $\mathcal{A}$ is *combinatorial* if two objects a, b in $\mathbb{A}$ are *isomorphic* if, and only if,

$$|\mathsf{hom}(c, a)| = |\mathsf{hom}(c, b)| \text{ for all } c \text{ in } \mathcal{A}$$

Theorem
Every locally finite category with *pushouts* and a *proper factorization* system iscombinatorial. **(D, Jakl, Reggio 2021)**

Pebble Games

The *pebbling* comonads were inspired by *pebble games*.

These are *model-comparison games* played on a pair of structures $\mathbb{A}$ and $\mathbb{B}$ by two players *Spoiler* and *Duplicator* using pebbles $a_1, \ldots, a_k$ on $\mathbb{A}$ and $b_1, \ldots, b_k$ on $\mathbb{B}$.
There is a *one-sided* version.

- *Spoiler* chooses a pair of pebbles a_i and b_i;
- *Duplicator* chooses a *function* $h : A \to B$ such that for pebbles a_j and $b_j (j \neq i)$, $h(a_j) = b_j$;
- *Spoiler* chooses $a \in A$ and places a_i on a and b_i on $h(a)$.

Duplicator loses if the partial map $a_i \mapsto b_i$ is not a partial *homorphism*.

Pebble Games

The *pebbling* comonads were inspired by *pebble games*.

These are *model-comparison games* played on a pair of structures $\mathbb{A}$ and $\mathbb{B}$ by two players *Spoiler* and *Duplicator* using pebbles $a_1, \ldots, a_k$ on $\mathbb{A}$ and $b_1, \ldots, b_k$ on $\mathbb{B}$.
There is a ~~one-sided~~ *two-sided* version.

- *Spoiler* chooses a pair of pebbles a_i and b_i;
- *Duplicator* chooses a ~~function~~ *bijection* $h : A \to B$ such that for pebbles a_j and $b_j (j \neq i)$, $h(a_j) = b_j$;
- *Spoiler* chooses $a \in A$ and places a_i on a and b_i on $h(a)$.

Duplicator loses if the partial map $a_i \mapsto b_i$ is not a partial ~~homorphism~~ *isomorphism*.

Notions of Width

variables in a conjunctive query
treewidth of graphs
pebbles in a one-sided game
} *Homomorphism*

variables in a counting logic formula
Weisfeiler-Leman dimension of graphs
pebbles in a two-sided bijection game
} *Isomorphism*

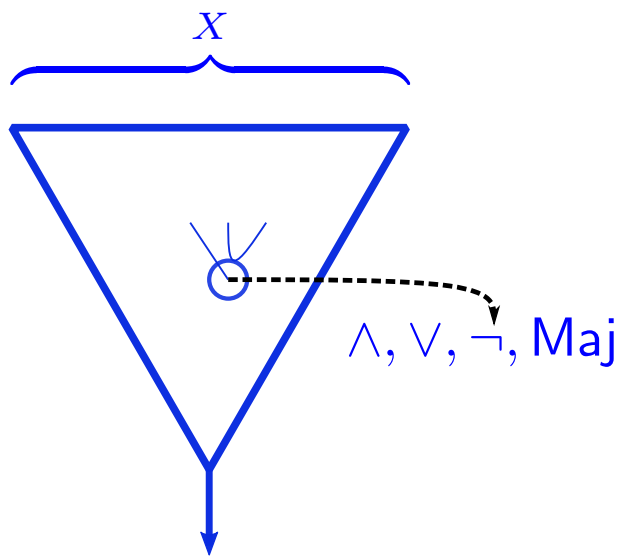
Circuits

A *language* $L \subseteq \{0,1\}^*$ can be described by a family of *Boolean functions*:

$$(f_n)_{n\in\omega} : \{0,1\}^n \to \{0,1\}.$$

Each f_n may be computed by a *circuit* C_n made up of

- Gates labeled by Boolean operators: $\wedge, \vee, \neg$,
- Boolean inputs: $x_1, \ldots, x_n$, and
- A distinguished gate determining the output.



Circuits for Graph Properties

We want to study families of circuits that decide properties of *graphs* (or other relational structures—for simplicity of presentation we restrict ourselves to graphs).

We have a family of Boolean circuits $(C_n)_{n\in\omega}$ where there are n^2 inputs labelled $(i,j) : i,j \in [n]$, corresponding to the *potential edges*.
Each input takes value 0 or 1;

Graph properties in P are given by such families where:

- the size of C_n is bounded by a polynomial $p(n)$; and
- the family is uniform, so the function $n \mapsto C_n$ is in P (or DLogTime).

Invariant Circuits

C_n is *invariant* if, for every input graph, the output is unchanged under a permutation of the inputs induced by a permutation of $[n]$.

That is, given any input $G : [n]^2 \to \{0, 1\}$, and a permutation $\pi \in S_n$,

C_n accepts G if, and only if, C_n accepts the input πG given

$$(\pi G)(i, j) = G(\pi(i), \pi(j)).$$

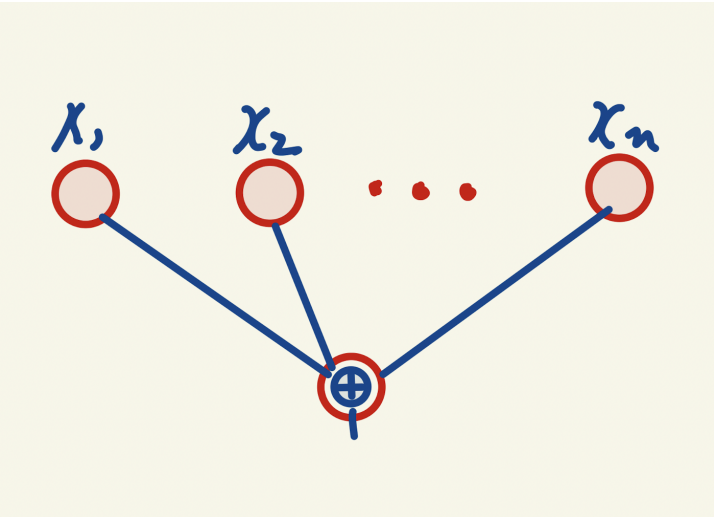
Say C_n is *symmetric* if any permutation of $[n]$ applied to its inputs can be extended to an automorphism of C_n.

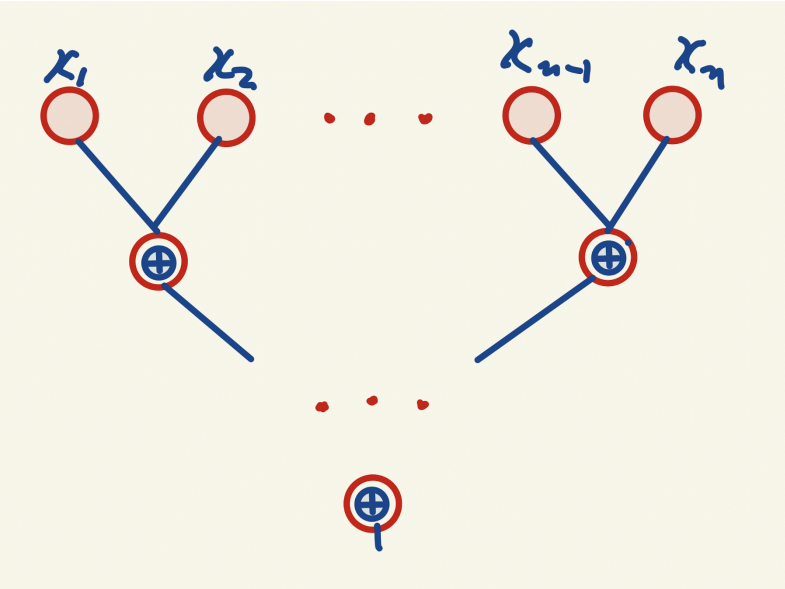
i.e., for each $\pi \in Sym_n$, there is an automorphism of C_n that takes input (i, j) to $(\pi i, \pi j)$.

Any symmetric circuit is invariant, but *not* conversely.

Symmetric Circuits

Example: $\bigoplus_{1 \leq i \leq n} x_i$.





Logic and Circuits

A graph property is definable in FPC *if, and only if,* it is decided by a P-uniform family of *symmetric* circuits using *AND*, *OR*, *NOT* and *MAJ* gates. (Anderson, D. 2017)

A graph property is definable in $C^{\omega}_{\infty\omega}$ *if, and only if,* it is decided by a family of *symmetric* circuits using *AND*, *OR*, *NOT* and *MAJ* gates with *polynomial size orbits*.

The *width* (number of variables) of a formula translates into a natural property of the circuits we call the *support size*.

Width and Supports

For a group $\Gamma \leq \mathsf{Sym}_n$, we say that a set $X \subseteq [n]$ is a *support* of Γ if

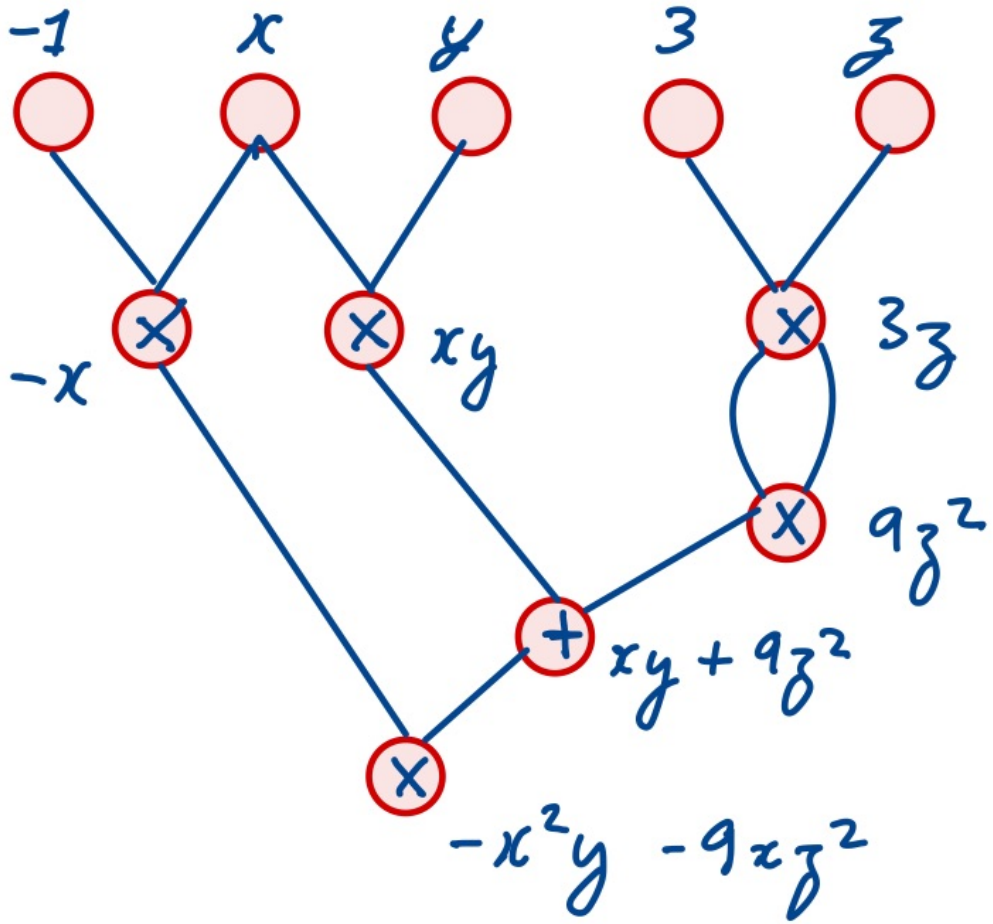
For every $\pi \in \mathsf{Sym}_n$, if $\pi(x) = x$ for all $x \in X$, then $\pi \in \Gamma$.

In other words, Γ contains all permutations that pointwise fix X.

Say that a circuit has *support size k* if every gate has a *stabilizer group* with a support of size at most k.

Circuit families with *bounded support size* have polynomial-size orbits and *vice versa*.

Algebraic Circuits



Algebraic Circuits

Algebraic Circuits over a field K are given by:

- A *directed acyclic graph*.
- Inputs labelled by a *variable* $x \in X$, or constant $c \in K$.
- Internal gates labelled by $+$ or $\times$.
- A designated *output*.

Each circuit computes (or represents) a *polynomial* in $K[X]$.

Valiant's conjecture VP $\neq$ VNP is the *algebraic analogue* of P $\neq$ NP.

Matrix Inputs

We are often interested in inputs which are entries of *a matrix*.

$$X = \{x_{ij} \mid 1 \leq i \leq m; 1 \leq j \leq n\}$$

Especially, when the input is a *square matrix*, so $m = n$.

$$\mathrm{Det}(X) = \sum_{\sigma \in \mathsf{Sym}_n} \mathsf{sgn}(\sigma) \prod_{i \in [n]} x_{i\sigma(i)}$$

$$\mathrm{Per}(X) = \sum_{\sigma \in \mathsf{Sym}_n} \prod_{i \in [n]} x_{i\sigma(i)}$$

Valiant's conjecture is equivalent to the statement $\mathrm{Per}(X)$ *cannot* be expressed by circuits of polynomial size.

Symmetric Algebraic Circuits

Suppose C is a circuit computing a polynomal $p \in K[X]$.

Sym_X—the group of *permutations* of X.

For $\Gamma \leq \mathsf{Sym}_X$, p is *Γ-symmetric* if for all $\pi \in \Gamma$, $p^\pi = p$.

C is *Γ-symmetric* if the action of Γ on the inputs X extends to an *automorphism* of C.

Symmetric Polynomials

The matrix polynomials $\mathrm{Det}(X)$ and $\mathrm{Per}(X)$ are both *square symmetric*, i.e. invariant under the action of Sym_n given by

$$x_{ij}^{\pi} = x_{\pi(i)\pi(j)}.$$

i.e., *simultaneous row and column permutations*.

$\mathrm{Per}(X)$ is also *matrix symmetric*, i.e. invariant under *independent row and column permutations*:
the action of $\mathsf{Sym}_n \times \mathsf{Sym}_n$ given by

$$x_{ij}^{(\sigma,\pi)} = x_{\sigma(i)\pi(j)}.$$

$\mathrm{Det}(X)$ is not matrix symmetric.

Polynomial-size Symmetric Circuits

There are *polynomial-size* square symmetric circuits for computing $\mathrm{Det}(X)$.

Any square symmetric circuits for computing $\mathrm{Per}(X)$ are of *exponential orbit size*.

(D, Wilsenach 2020)

We can, in fact, get a complete characterization of *matrix symmetric* polynomials that are computable by symmetric circuits with polynomial orbit size.

Homomorphism Polynomials

An $a \times b$ matrix can be seen as a *weighted bipartite graph* on the set of vertices $[a] \uplus [b]$.

Given such a matrix M and the matrix of variables $X = \{x_{ij} \mid i \in [m], j \in [n]\}$, we define the *homomorphism polynomial of* M as:

$$\mathsf{hom}_{M,m,n} = \sum_{f:[a]\to[m], g:[b]\to[n]} \prod_{i\in[a], j\in[b]} M_{i,j} x_{f(i),g(j)}.$$

When M is the biadjacency matrix of a graph F, this gives an expression such that substituting for X the biadjaceny matrix of a graph G, evaluating to the *number of homomorphisms* from F to G.

Characterization

Theorem

A family of polynomials $p_{m,n}(X)$ is computed by a family of *matrix symmetric* circuits of *polynomial circuit size*

if, and only if,

there is a k such that each $p_{m,n}$ is a linear combination of homomorphism polynomials $\hom_{M,m,n}$ of graphs of treewidth less than k.

(D,Pago,Seppelt 2025)

Conclusions

How many variables are needed to express a formula?

This is a measure of *complexity* that ties in to many *natural* and *independently discovered* measures in combinatorics, logic and complexity.

Proving lower bounds yields *surprising* and *unconditional* lower bounds in circuit complexity and other areas.

Closely related notions of width arise in other areas, e.g. *constraint satisfaction problems*; *submodular width* ...