Counting Complexity

Examples: #SAT, #CLIQUES, #PERFECT_MATCHINGS....

Definition: $f: \Sigma^* \to \mathbb{N}$ is in #P if there exists $A \in P$ and a polynomial p such that:

$$\forall x \in \Sigma^n \quad f(x) = \#\{ y \in \Sigma^{\leq p(n)} \mid \langle x, y \rangle \in A \}.$$

This is the counting analogue of NP (Valiant '79).

The permanent: If A is an $n \times n$ matrix,

$$\text{perm}(A) = \sum_{\sigma \in S_n} a_{1\sigma(1)} a_{2\sigma(2)} \cdots a_{n\sigma(n)}.$$

perm is in #P for matrices with entries in $\mathbb{N}$.

$$\text{perm}(A) = \#\left\{(\sigma, y);\ 1 \le y \le a_{1\sigma(1)} a_{2\sigma(2)} \cdots a_{n\sigma(n)}\right\} \quad (1')$$

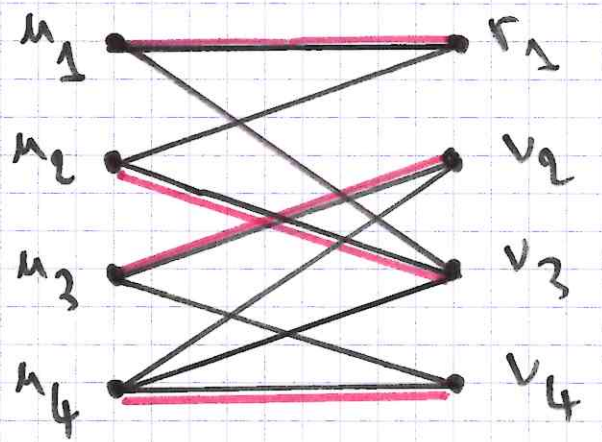
can be checked in polynomial time

$$\mathrm{perm}(A) = \#\{(\sigma, y) : 1 \le a_{1\sigma(1)} a_{2\sigma(2)} \cdots a_{n\sigma(n)} \le y\}$$

can be decided in polynomial time

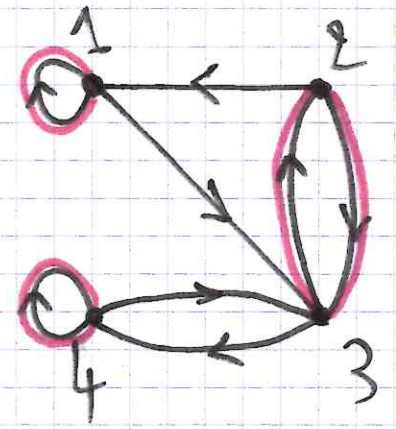
Combinatorial interpretation: Suppose $A \in \{0,1\}^{n^2}$.

Perfect matchings in bipartite graphs:



$\sigma: 1 \mapsto 1, 2 \mapsto 3, 3 \mapsto 2, 4 \mapsto 4$

Cycle covers in directed graphs:



Remark: $\#P \subseteq FPSPACE$. $\#P$ is harder than NP, and in fact harder than PH (Toda's theorem).

#P-completeness: $f \in \#P$ is #P-complete if for all $g \in \#P$, there exists 2 polytime computable functions I, O such that:

$$g(x) = O(f(I(x)))$$

Sometimes: several calls to f in the reduction

If O is the identity function, the reduction is **parsimonious**.

Most reductions between NP-complete problems are parsimonious (or can be made parsimonious)

$\Rightarrow$ #SAT, #HAMILTONIAN_PATH, ... are #P-complete for parsimonious reductions.

Is #PERFECT_MATCHINGS #P-complete?

The reduction cannot be parsimonious.
Otherwise, $\#SAT(\varphi) = \#PERFECT\_MATCHINGS(I(\varphi))$

φ: SAT instance

$I(\varphi)$: graph constructed from φ.

We can decide whether this is ≥ 1 in polynomial time!

Theorem: counting perfect matchings in bipartite graphs is #P-complete.

The proof involves the permanent of matrices with entries in $\mathbb{Z}$.

Let's take a look:

Modular Counting

In the reduction we construct a matrix A such that:

$\text{perm}(A) = 4^m s$.

s ≈ # of satisfiable assignments.

So $\text{perm}(A) \bmod 3 = s \bmod 3$:

permanent mod 3 is hard (at least as hard as Mod_3P).

permanent mod 2 is easy.

at least as hard as NP

(Valiant - Vazirani)

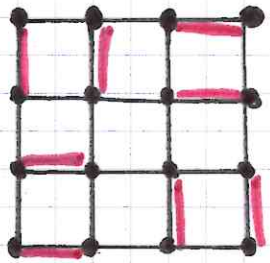
Some efficient algorithms

The permanent is polytime computable for matrices:

- of bounded rank (Barvinok '96).
- of bounded treewidth (Courcelle-Makowsky-Rotics '01): view A is the adjacency matrix of a graph.
- planar graphs (Fisher-Kasteleyn-Temperley, '60's):

what about other widths?

$$\#\text{PERFECT\_MATCHINGS}(G) = \sqrt{\det(A)}$$ where A is a "Pfaffian orientation" of G.



A "dimer covering"

FKT is the engine behind Valiant's « holographic algorithms » (2007)

Approximation algorithms for perm

For non negative matrices:

- Markov chains (Jerrum - Sinclair).
- matrix scaling (Linial - Samorodnitsky - Wigderson, deterministic)

Average-case complexity of the permanent

It is as hard as the worst case (Lipton '91)
in large enough finite fields (or bounded integer entries)

What about $\mathbb{F}_3$?

What about 0/1 entries ?

Open problem in « Counting, sampling and integrating: algorithms and complexity » (2003) by Mark Jerrum

The hardness proof breaks because it relies on polynomial interpolation: to evaluate per(A), interpolate f(x)=per(A+xR) where R is a random matrix. More one the permanent polynomial in the next slides!

Algebraic Complexity

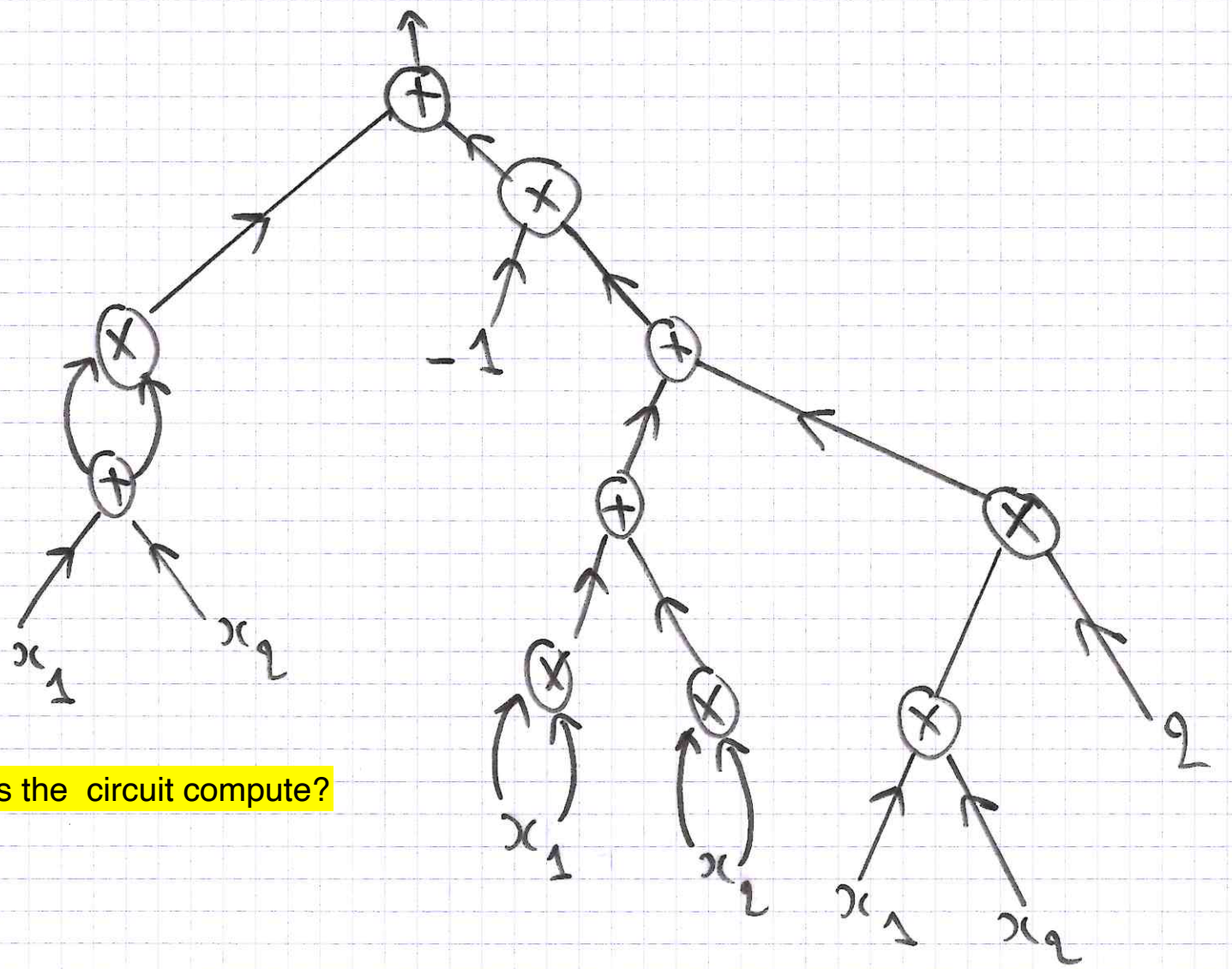
perm is a degree n polynomial in n^2 variables.

Can we compute it in poly(n) arithmetic ops $(+, \times)$?

Can we prove it?

For this we need a computation model.

Arithmetic Circuits



What does the circuit compute?

VP

A polynomial family (f_n) is in VP if:

1. f_n has poly(n) variables.
2. $\deg(f_n) \leq \text{poly}(n)$.
3. f_n can be computed by an arithmetic circuit C_n of polynomial size.

Example: The determinant (not quite obvious).

VNP

An algebraic analogue of #P !

(f_m) is in VNP if $f_m(\bar{x}) = \sum_{\bar{y} \in \{0,1\}^{r(m)}} g_m(\bar{x}, \bar{y})$

for some family (g_m) in VP.

Example: The permanent.

Theorem: perm is VNP-complete in any field K with $\mathrm{char}(K) \neq 2$.

If (f_m) is in VNP, f_m can be written as a poly(m) size permanent.

The entries are variables and constants (1/2 is used).

Example: $\mathrm{perm} \begin{pmatrix} x_1 & 0 & x_2 \\ 1/2 & x_1 & 1 \\ x_1 & x_2 & x_2 \end{pmatrix}$

Limaye - Srinivasan - Tavenas'22: superpolynomial lower bound for constant-depth arithmetic circuits.