October 14, 2026

Admin

- Typos in HW 2, problem 4(b) & 4(d). Deadline extended 48 hours.
- HW 3 out. Due in 2 weeks.
- Start thinking about course project. Lightweight proposal due in 2 weeks.

Today: Start arithmetic circuits & complexity theory.

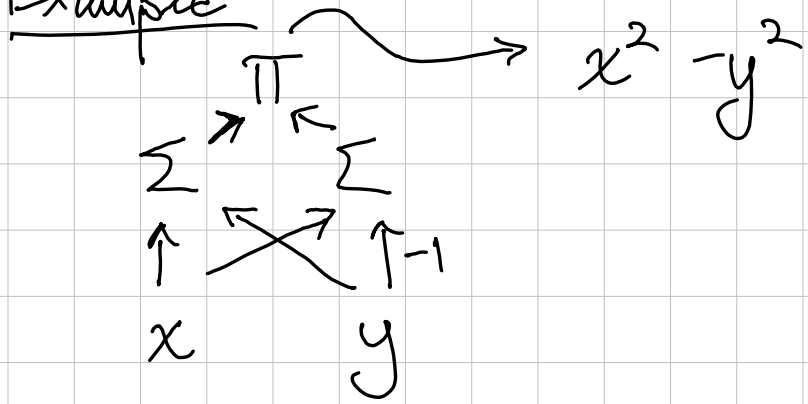
So far, we have studied algorithms for a variety of problems. Now, we turn to relationships between problems & the study of computational intractability.

We will study arithmetic circuits, a natural and well-studied model of computation for multivariate polynomials.

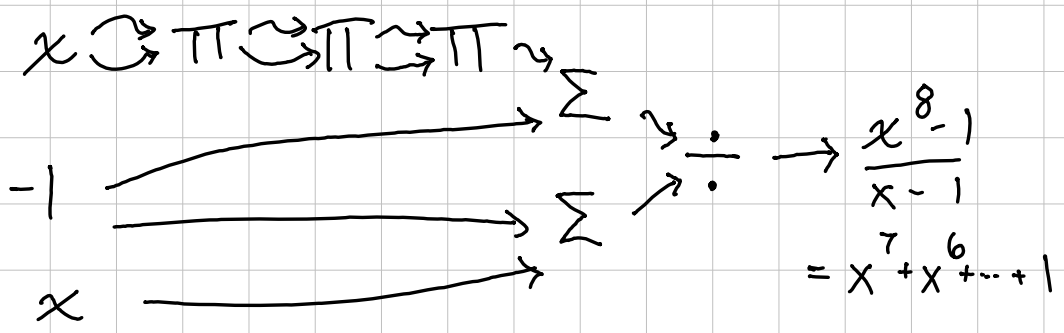
Definition An arithmetic circuit C is a directed acyclic graph G. Vertices of G are called gates and edges are called wires. Each gate of fan-in zero is labeled by a variable or a constant $\alpha \in \mathbb{C}$. Internal gates are labeled by arithmetic operations $(+, \times, \div)$. Every gate computes a rational function in the natural way. Gates of fan-out zero are called output gates, and the circuit is said to compute the rational functions computed by the output gates.

The size of the circuit is the # of gates. The depth is the length of the longest input-output path.

Example



Example



We will focus on computation of families of multivariate polynomials of low degree.

Definition A family $f = (f_n)_{n \in \mathbb{N}}$ of polynomials is a p-family if

- f_n is a polynomial in $m(n) \leq \text{poly}(n)$ variables
- $\deg(f_n) \leq \text{poly}(n)$.

Definition A p-family (f_n) is p-computable if f_n can be computed by arithmetic circuits of size $\text{poly}(n)$

$$VP := \left\{ (f_n) : (f_n) \text{ is p-computable} \right\}$$

VP is analogous to P. The analogue of NP is VNP.

Definition A p-family (f_n) is p-definable if $\exists\, (g_n) \in VP$ st

$$f_n(x_1, \ldots, x_m) = \sum_{e \in \{0,1\}^{m'}} g(x_1, \ldots, x_m, e_1, \ldots, e_{m'}).$$

$$VNP = \{(f_n) : (f_n) \text{ p-definable}\}$$

What about NP-completeness & reductions?

<u>Definition</u> Let f, g p-families. Say f is a p-projection of g if

$$f_n(x_1, \ldots, x_m) = g_{t(n)}(a_1, \ldots, a_{v(n)})$$

for $t(n) \leq \text{poly}(n)$

$a_1, \ldots, a_{v(n)} \in \mathbb{C} \cup \{x_1, \ldots, x_m\}$.

<u>Theorem</u> (Valiant) The permanent

$$\text{perm}_n(X) = \sum_{\sigma \in S_n} \prod_{i=1}^{n} x_{i,\sigma(i)}$$

is VNP-complete under p-projections.

The determinant

$$\det_n(X) = \sum_{\sigma \in S_n} \operatorname{sgn}(\sigma) \prod_{i=1}^{n} x_{i,\sigma(i)}$$

(almost) plays the role of a complete problem for VP.

We defined arithmetic circuits to allow divisions. Polynomials can be computed without divisions. Do they add any power?

No!

Theorem (Strassen) If $f = (f_n)$ is in VP, then f_n can be computed by division-free arithmetic circuits of size poly(n).

Idea: Push division to the top of the circuit & eliminate it.

Each gate v computes a rational function f_v / g_v. We'll split v into two gates (v, num) and (v, den) that are intended to compute the numerator f_v & denominator g_v.

- If $v = x_i$ or $v = a \in \mathbb{C}$ is an input, set
  $(v, num) = x_i$ or a
  $(v, den) = 1$

- If $v = u \times w$ is a multiplication gate, set
  $(v, num) = (u, num) \times (w, num)$
  $(v, den) = (u, den) \times (w, den)$

- If $v = u / w$ is a division gate, set
  $(v, num) = (u, num) \times (w, den)$
  $(v, den) = (u, den) \times (w, num)$

- If $v = u + w$ is an addition gate, set

$$(v, num) = (u, num) \times (w, den) + (u, den) \times (w, num)$$
$$(v, den) = (u, den) \times (w, den)$$

The resulting circuit uses no divisions and only increases in size by a constant factor.

At the top of the circuit, we compute two polynomials $f(x)$, $g(x)$ such that $f(x)/g(x)$ is the polynomial computed by the original circuit.

To perform this division without dividing, we invert g as a power series.

Recall $\frac{1}{1-x} = 1 + x + x^2 + \cdots$ as power series.

If $g(\vec{0}) \neq 0$, write

$$g(\vec{x}) = g(\vec{0}) - \hat{g}(\vec{x}).$$

Then

$$\frac{f}{g} = \frac{f}{g(0) - \hat{g}} = \frac{f(\vec{x})}{g(\vec{0})} \sum_{i \geq 0} \left(\frac{\hat{g}(\vec{x})}{g(\vec{0})} \right)^i.$$

We'll see next time how to turn this into a circuit that computes $f(\vec{x}) / g(\vec{x})$.