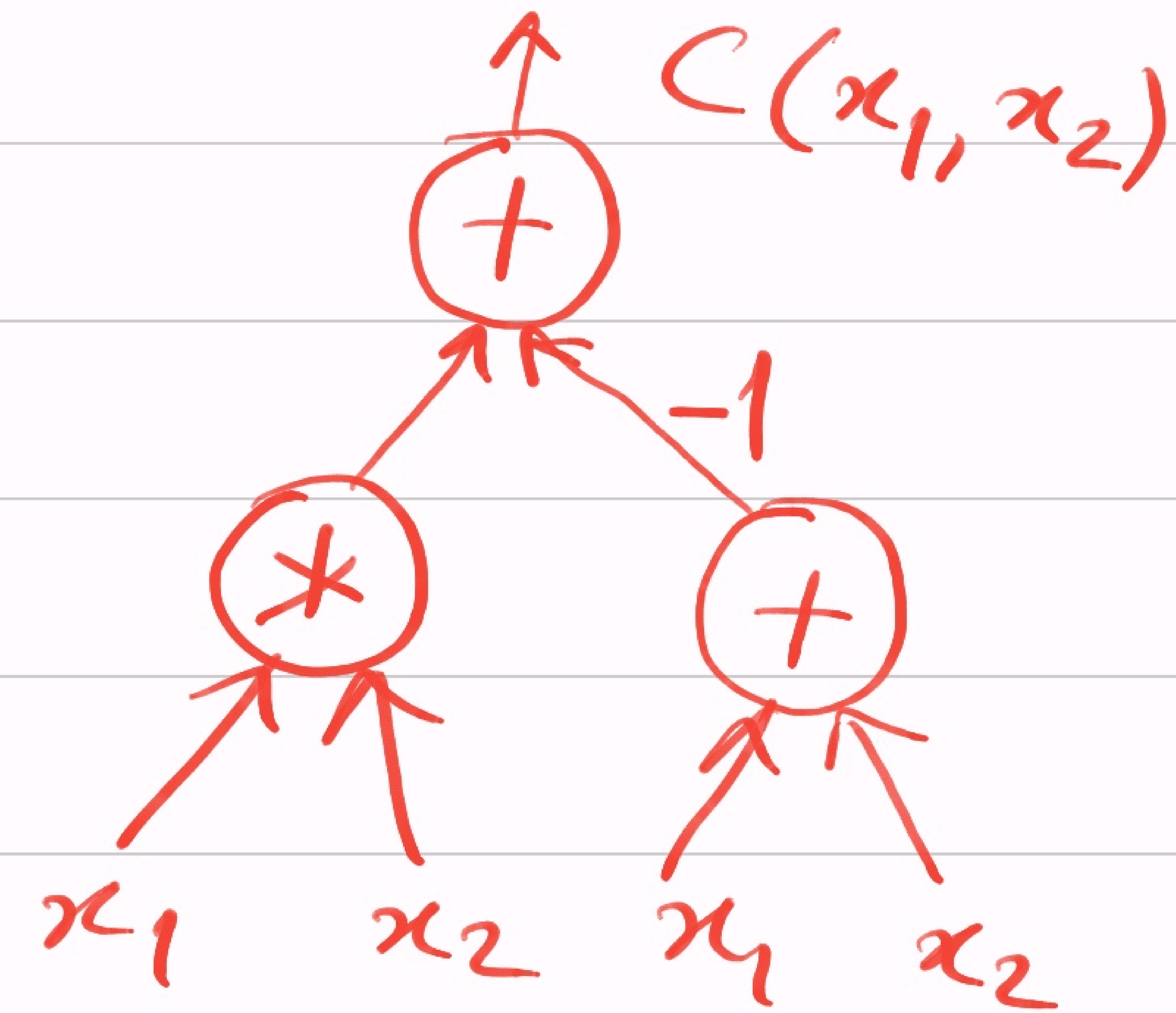
- Arithmetic circuit: A rooted tree with inputs as leaves, output as root, $+/*$ as nodes, & constants on edges.



- Size = #edges + #nodes + bit-size(constants)

▷ Circuit can capture very large polynomials, in small-size!

— Eg. $(x_1 + \ldots + x_n)^n$ → $\approx 2^n$ monomials; → $O(n)$-size

▷ PIT is nontrivial. We want poly(size)-time.

<u>Theorem:</u> $PIT \in BPP$.

Pf: • Idea: Evaluate circuit at random point.

• Let $C(\bar{x})$ be the given circuit, over $\mathbb{F}$, of size s.

▷ $\deg C(\bar{x}) \le s^s$. [Each $*$-gate could grow the degree s-times.]

• Pick a subset $S \subset \mathbb{F}$ s.t. $|S| > 2 \cdot s^s$.

[If $\mathbb{F}$ is small then go to a field extension.]

• Algorithm samples from S:

1) Pick random point $\underbrace{(a_1, \ldots, a_n)}_{}$ $=: \bar{a} \in S^n$.

2) If $C(\bar{a}) = 0$ then OUTPUT zero else OUTPUT nonzero.

- **Correctness:**

▷ $C(\bar{x}) = 0 \Rightarrow \text{Prob}[\text{correct output}] = 1$.

▷ $C(\bar{x}) \neq 0 \Rightarrow \text{Prob}[\text{correct output}]$

[P.I. lemma] $\geq 1 - \frac{\delta^{\delta}}{|S|} = \frac{1}{2}$.

▷ Error prob. can be reduced by repeating. $\square$

P.I. lemma (DeMillo-Lipton '78, Zippel '79, Schwartz '80):
Let $P \in \mathbb{F}[\bar{x}]$ be a polynomial of degree $d \geq 0$.
Then, $\Pr_{\bar{a} \in S^n}[P(\bar{a}) = 0] \leq d/|S|$.

Proof: • For $n=1$, it follows from the fact that $P(x_1)$ has $\leq d$ roots in $\mathbb{F}$.

• Let's induct on n:

• Assume it to be true for $(n-1)$-variables.

• Write $P =: \sum_{0 \leq i \leq d} x_n^i \cdot \underline{P_i(x_1, \ldots, x_{n-1})}$.

• As $P \neq 0$, let i_0 be the largest i s.t. $P_{i_0} \neq 0$.

$$\Rightarrow \Pr[P(\bar{a})=0] = \Pr[P_i(\bar{a})=0] \cdot \Pr[P(\bar{a})=0 \mid P_i(\bar{a})=0]$$
$$+ \Pr[P_i(\bar{a}) \neq 0] \cdot \Pr[P(\bar{a})=0 \mid P_i(\bar{a}) \neq 0]$$

$$\leq \Pr[P_i(\bar{a})=0] + \Pr[P(\bar{a})=0 \mid P_i(\bar{a}) \neq 0] \leq \frac{d-i_0}{|S|} + \frac{i_0}{|S|}$$

$$\leq d/|S|.$$

$\square$ induction

- <u>Exercise</u>: How do you construct extension of $\mathbb{F}$?
[do it in poly(s)-time?]

The Circuit Model

- An arithmetic circuit (over $\mathbb{F}$) has $+, *$ gates & field elements.
  $\Rightarrow$ computes polynomials.
- A boolean circuit has $\wedge, \vee, \sim$ gates & constants $= \{0,1\}$ or $\{False, True\}$.
  $\Rightarrow$ computes boolean formula.
- We can use these as a model of computation instead of TMs.

- **Defn:** • A problem $L \subseteq \{0,1\}^*$ is said to be solved by a boolean circuit family $\{C_n(\bar{x}) \mid n \geq 1\}$ if $\forall n$, $\forall x \in \{0,1\}^n$, $C_n(x) = 1$ iff $x \in L$.

  • Computational resources are: $size(C_n)$, $depth(C_n)$ & $fanin/fanout(C_n)$.

- You can prove the following simple facts:

Proposition: 1) Any TM can be turned into a boolean circuit family. [vice versa?]

2) Boolean circuits are inspired from "electronics" & capture parallel computation :

size(C) $\rightsquigarrow$ space of the parallel algorithm.

depth(C) $\rightsquigarrow$ time " " " " .

3) Two n-bit integers can be added by a poly(n)-size & constant-depth boolean circuit.

[Exercise: What about multiplying two n-bit integers?]

Circuit Complexity Classes

- Analogous to $Dtime(T(n))$ we have,

  $Size(s(n)) := \{ L \subseteq \{0,1\}^* \mid \exists\ O(s(n))\text{-size}$ boolean circuits $\{C_n\}$ solving $L \}$.

- $P/poly := \bigcup_{c>0} Size(n^c)$.

- Analogously, for arithmetic circuits :

  $Ar\text{-}size(s(n)) := \{ \{f_n\}_{n\geq 1} \mid \exists\ O(s(n))\text{-size}$ arithmetic circuits $\{C_n\}_n$ s.t. $C_n = f_n \}$

- $Ar\text{-}P/poly := \bigcup_{c>0} Ar\text{-}Size(n^c)$.

▷ $\begin{cases} \text{P/poly has boolean functions } f_n : \{0,1\}^* \to \{0,1\} \\ \text{Ar-P/poly has polynomial } f_n : \mathbb{F}^n \to \mathbb{F}. \end{cases}$

▷ $P \subsetneq P/poly$

- But Ar-P/poly is incomparable with P.