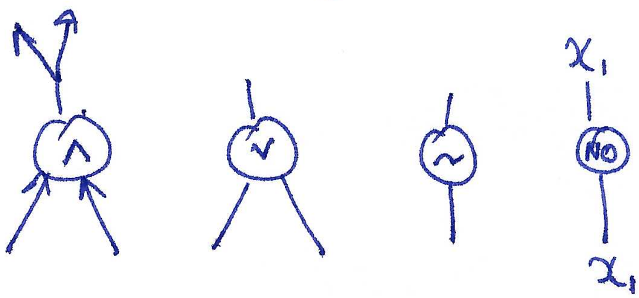
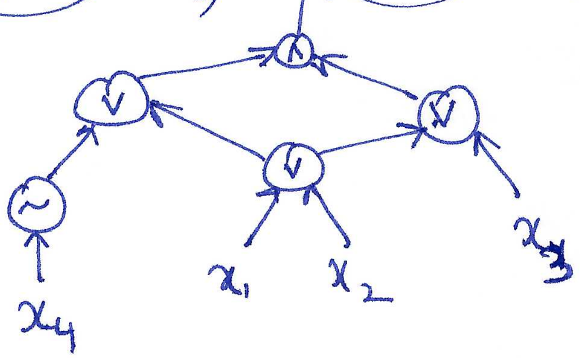
Circuit Complexity

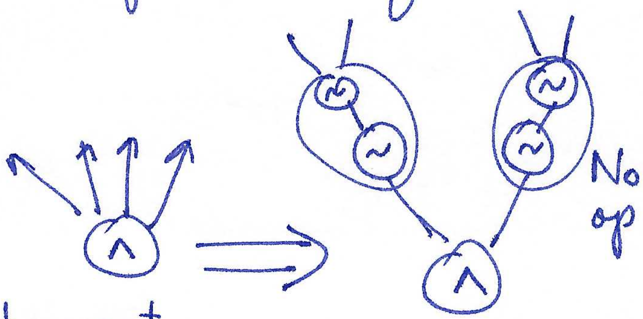


Def: A circuit is a DAG, where leaves/sources are labeled $\underbrace{x_1, \dots, x_n}_{\text{inputs}}$, or 0, 1. Internal nodes are labeled $\vee, \wedge, \sim$, and each gate has outdegree/indegree at most 2.

If outdegree is always 1, then the circuit is a tree (formula).
Each x_i appears in EXACTLY one leaf.

$f(\cdot\cdot) = ((x_1 \vee x_2) \vee x_3) \wedge ((x_1 \vee x_2) \vee \bar{x}_4)$





fan-out $\Rightarrow$ binary tree that "copies" value

The size of a circuit is the number of vertices ~~+ edges~~ in the DAG.

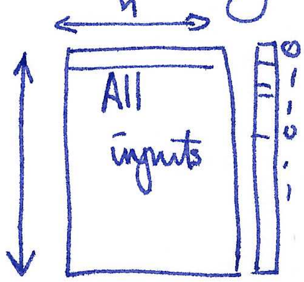
In our setting, size $= \cancel{\Theta(\#\text{ vertices})} = \cancel{\Theta(\#\text{ gates})}$

A circuit C_n (on n inputs) computes a fn $f: \{0,1\}^n \to \{0,1\}$ if $\forall\, x$ of length n, $C_n(x) = f(x)$.

<u>Thm [Shannon 49]</u> [Lupanov] For every n, there exists a function f_n that requires a circuit of size $\Omega\left(\frac{2^n}{n}\right)$ to compute it.

Moreover, for ALL functions $f:\{0,1\}^n \to \{0,1\}$, f can be computed by a circuit of size $O\left(\frac{2^n}{n}\right)$.

<u>Proof</u>: How many functions $f:\{0,1\}^n \to \{0,1\}$ exists?



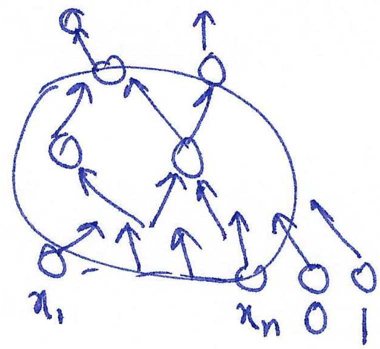
2 choices $f(x) \to \{0,1\}$

How many possible x's?

$$2^{2^n}$$

Let us count the number of circuits of size s (s gates)

Let us represent circuit by adjacency list. The leaves are numbered $x_1, \ldots x_n; 1, 0$.



The representation has to specify the gate at each vertex, and the outneighbors each gate. The leaves need a label in $x_1, \ldots, x_n, 1, 0$

different circuits of size $s \leq$ (# ways of labeling leaves) $\times$ (# ways of setting outneighbors) $\times$ (# ways of setting gates)

$$\leq (n+2)! \times \binom{s}{2}^s \times 3^s \leq s^{2s} \times s^{2s} \times 3^s = 2^{[4s\lg s + s\lg 3]}$$

$$2^{[4s\lg s + s\lg 3]} \leqslant 2^{[5s\lg s]} < 2^{2^n}$$

Suppose $s < \dfrac{2^n}{\underbrace{c}_{\text{constant} > 5}\,\cancel{\lg}\, n}$

$$5s\lg s < \frac{5\times 2^n}{c\times n} \times \lg\left(\frac{2^n}{cn}\right)$$

$$\leqslant \frac{5\times 2^n}{c\times \cancel{n}} \times \cancel{n} \leqslant 2^n$$

The number of distinct circuits of size $< \dfrac{2^n}{5n}$ is strictly smaller than 2^{2^n}.

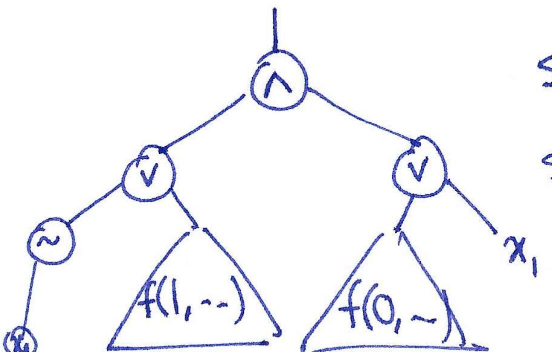
Thus, there exists a function $f_n : \{0,1\}^n \to \{0,1\}$ that requires a circuit of size $\geq \dfrac{2^n}{5n}$ to compute it.

<u>Clm</u>: Consider any $f : \{0,1\}^n \to \{0,1\}$. There exists a circuit of size 50×2^n that computes f.

<u>Proof</u>: Proof by induction.

$$f(x_1, \text{---} x_n) = (\sim x_1 \vee f(1, x_2 \text{---}, x_n)) \wedge (x_1 \vee f(0, x_2 \text{--}, x_n))$$

(If $x_1 = 1$, $f(1, \text{–} x_n)$) AND (If $x_1 = 0$, $f(0, \text{--} x_n)$)

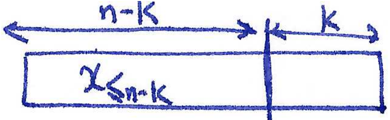


$$\text{size}(n) \leqslant 2 \times \text{size}(n-1) + 10$$

$$\text{size}(n) \leqslant 10 \sum_{i=1}^{n} 2^i = \cancel{\Theta(}$$

$$\leqslant 50 \times 2^n$$

Prove by induction



$x = (x_1, \ldots, x_n)$

$$x = (x_{\leq n-k}, \underbrace{x_{n-k+1}, x_{n-k+2}, \ldots, x_n}_{k \text{ bits}})$$

If we fix $x_{\leq n-k}$ to (say) z, $f(\underset{\text{fixed}}{z}, \underbrace{x_{n-k+1}, \ldots, x_n}_{k \text{ bits}})$

we get a function $f_z : \{0,1\}^k \to \{0,1\}$.

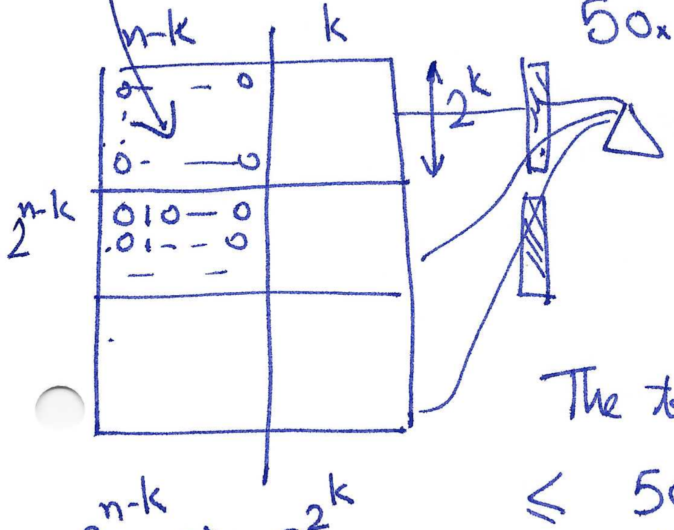
Think of f as collection of $\underset{2^{n-k}}{\ } f_z$'s

There are 2^{2^k} possible f_z's. Suppose we "precompute" all of them using a separate circuit.

Build this using previous claim

$\underbrace{\text{circuit for each possible } f_z}_{50 \times 2^{(2^k+k)}}$ $\quad 2^{2^k} \times 50 \times 2^k$

n-k bits are same



Follow previous clm construction for (n-k) bits and then use the f_z circuits

The total size

$$\leq 50 \times 2^{n-k} + 50 \times 2^{(2^k+k)}$$

$$= O\left(\frac{2^n}{2^k} + 2^{(2^k+k)}\right)$$

$$2^{n-k} \gg 2^{2^k}$$

Choose k so that the size is $O\left(\frac{2^n}{n}\right)$.

<u>Def</u>: A circuit family is a collection of circuits $\mathcal{C} = \{C_n\}_{n \in \mathbb{N}}$ where C_n has n input bits.

A circuit family $\mathcal{C}$ computes/decides language $\mathcal{L}$ if $\mathcal{L} = \{x \mid C_n(x) = 1 \text{ where } x \text{ has length } n\}$

~~$\mathcal{L}$ is~~ A circuit family $\mathcal{C}$ has size $s(n)$ if
$\forall n, \ \text{size}(C_n) = O(s(n))$

<u>Thm</u>: For <u>all</u> languages $\mathcal{L}$, $\mathcal{L}$ is computed by a circuit family of size $\frac{2^n}{n}$.

Even undecidable $\mathcal{L}$!

The description of circuit family $\mathcal{C}$ is potentially infinite. <u>An algorithm/TM always has a finite description.</u>

Non-uniformity model of comp. : potentially infinite description, "different" algorithm for every $n \leftarrow$ input size

Uniform model of comp : finite description (TM/algorithm)

Def: $\mathbb{SIZE}(s(n))$ is the family of languages computed by $s(n)$-sized circuit families.

Circuit complexity

All languages lie in $\mathbb{SIZE}\left(\frac{2^n}{n}\right)$.

$$\mathbb{P}/\text{poly} = \bigcup_{k \in \mathbb{N}} \mathbb{SIZE}(n^k)$$

$\hookrightarrow$ all languages computable by polynomial sized circuits.

If $L \in \mathbb{P}/\text{poly}$, we can construct "efficient" circuits to decide L.

Thm: [Non-uniform hierarchy theorem] $\left(\text{DTIME}(t(n)) \subsetneq \text{DTIME}(t'(n)) \text{ when } t'(n) > t(n)\log t(n)\right)$

Let $n < s(n) < \frac{2^n}{n}$. Then $\mathbb{SIZE}(s(n)) \subsetneq \mathbb{SIZE}(4s(n))$

Easy consequence of Shannon's theorem.