$f: \{0,1\}^n \to \{0,1\}$

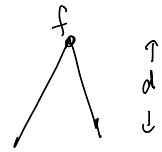
$f: \{0,1\}^* \to \{0,1\}$

$C_1, C_2, \ldots$

f has depth $O(\log n)$

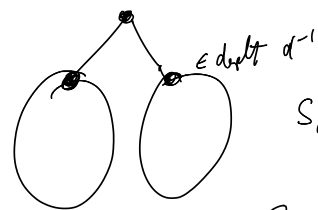
$\Leftrightarrow$ For every n, $\exists$ circuit C_n of depth $O(\log n)$ computing f on inputs of length n.

Size vs depth of circuits



$\longrightarrow$ size $\le O(2^d)$

Proof by induction



$S_d \le 2 \cdot S_{d-1} + 1$

$\vdots$

$S_d \le O(2^d)$

Open problem 1

Show that some explicit function cannot be computed simultaneously in $O(\log n)$ depth and $O(n)$ size.

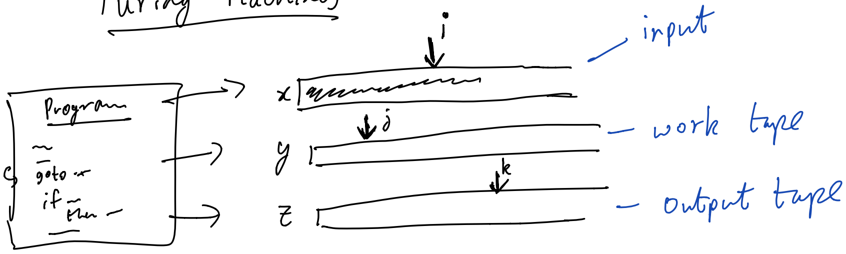
Open problem 2

Can every circuit of size $\mathrm{poly}(n)$ be simulated by a circuit of depth $O(\log n)$?

Open problem 3

Show that for some explicit function there is no $O(1)$ width program of length $O(n \log^3 n)$ computing.

Turing Machines



input: read only
work tape: read/write
output: write only

(uniform model)

Resources

Space: max value of j reqd. during computation.

time: # of steps in computations.

Fact: space $\le$ time.

(Extended) Church-Turing Thesis: Everything that can be computed (efficiently) can be computed (efficiently) by a T.M.

Claim: T.M with alphabet size L, time $T(n)$ $\rightarrow$ T.M with binary alphabet, running time $O(\log L \cdot T(n))$

Claim: T.M with L tapes, time $T(n)$ $\rightarrow$ T.M with 3-tapes, time $O(L \cdot T(n)^2)$

Thm: There is a T.M. M such that given code α, input x, if α takes T steps to compute on x, then $M(\alpha, x)$ computes same output in $O(C\, T \log T)$ steps, where C is determined by α.

time $T(n)$ $\longrightarrow$ size $O(T \log T)$