Fri April 20, 2018 - page 1

Thursday, April 19, 2018 2:04 PM

New model of computation: circuits (instead of the Turing Machine)

Advantages of using circuits:
- Enables discussion of parallelism
- More manageable lower bounds.
- Introduces new ideas about uniformity + advice

Circuit C
- directed acyclic graph
- nodes labeled with

$\wedge$ $\vee$ (in-degree = 2) $\neg$ (in-degree = 1) x (variable) 0 1 (in-degree = 0.)

- n = # of input variables.
- there is one "sink" (node of out-degree = 0) → output of the circuit.

A circuit computes a function $f: \{0,1\}^n \to \{0,1\}$
$C \Leftrightarrow f$.

Complexity of a circuit: # of gates.

Also depth: length of the longest path from an input to the output.

A **formula** is a ckt in which the graph is a tree (out-degree of all gates = 1, except for the output)

Thursday, April 19, 2018 2:04 PM

Every function $f: \{0,1\}^n \to \{0,1\}$ is computable by a circuit of size $O(n2^n)$

Take the AND of n literals that encode x s.t. $f(x) = 1$ then OR the $\leq 2^n$ such terms.

x_1	x_2	x_3	$f(x_1x_2x_3)$	
0	0	0	0	
0	0	1	0	
0	1	0	1	$\sim (\neg x_1 \wedge x_2 \wedge \neg x_3) \vee$
0	1	1	0	
→ 1	0	0	1	$\sim (x_1 \wedge \neg x_2 \wedge \neg x_3) \vee$
1	0	1	1	$\sim (x_1 \wedge \neg x_2 \wedge x_3$
1	1	0	0	
1	1	1	0	

Note: Circuits only work for a specific input size.
We're used to TMs which compute $f: \Sigma^* \to \{0,1\}$

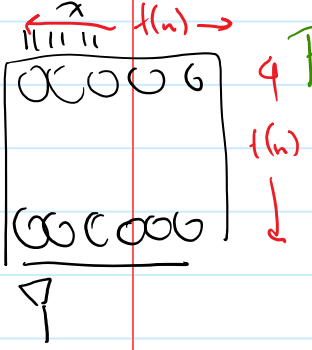
Circuit Families: a circuit for each input length.
C_1 C_2 C_3 ...
$\{C_n\}$ computes $f: \Sigma^* \to \{0,1\}$ iff $\forall x \in \Sigma^*$
$C_{|x|}(x) = f(x)$

We say that $\{C_n\}$ decides L:
$x \in L \iff C_{|x|}(x) = 1.$

Thursday, April 19, 2018 2:04 PM

How do the circuit and Turing Machine model compare?

($\Rightarrow$) If there is a TM that decides L in time $t(n)$ then there is a circuit family that decides L, where the size of C_n is $O(t(n)^2)$.



Proof: CVAL construction

Tableau representing TM computation on variable x.

Can be turned into a circuit of size $O(t(n)^2)$.

$\Leftarrow$? If there is a circuit family that computes L, then can L be computed by a TM?

Example: $C_n = (x \vee \neg x)$ if M_n halts.

$C_n = (x \wedge \neg x)$ if M_n loops.

This circuit family decides a unary version of the halting problem

by encoding uncomputable information into the specifications of the circuit family.

Solution: Uniformity.

Require that the specifications of the circuit be easy to compute.

Thursday, April 19, 2018 2:04 PM

Definition: A circuit family $\{C_n\}$ is logspace uniform iff $\exists$ TM which outputs C_n on input 1^n and runs in log space.

Theorem: P = languages decidable by a logspace uniform poly-sized circuit family.

(P defined for TM $\neq$ circuit)

$P \Rightarrow \{C_n\}$ (done earlier)

$C_1\ C_2 \ldots C_n \ldots$

gates in $C_n \leq$ poly(n).

$\{C_n\} \Rightarrow$ poly-time TM

M generates $C_{|x|}$
then evaluates $C_{|x|}(x)$
accept iff $C_{|x|}(x) = 1$.

Turing Machines with Advice

A circuit family without the uniformity constraint is called "non-uniform".

We can regard non-uniformity like another resource like space and time.

→ Add read-only advice tape to TM M.

→ Advice only depends on n, the size of the input ($A(n)$ = advice).

→ M decides L w/ advice $A(n)$ iff $M(x, A(|x|))$ accepts $\iff x \in L$

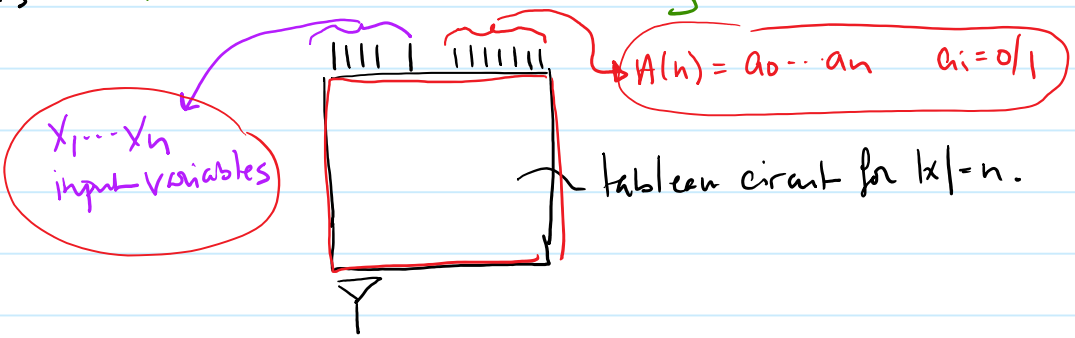
Thursday, April 19, 2018 2:04 PM

Complexity class: $TIME(t(n))/f(n)$
= the set of languages for which
$\exists A(n)$ $A: \mathbb{N} \to \Sigma^*$ $|A(n)| \leq f(n)$
there is a TM M that decides L in time $t(n)$ with advice A.

Most important advice class: $P/poly = \bigcup_{k \geq 1} TIME(n^k)/n^k$.

Theorem: $L \in P/poly$ iff L is decided by a family of (non-uniform) poly-sized circuits.

$P/poly \to \{C_n\}$ Hard-code the advice string into the circuit:



$\{C_n\} \to P/poly$ $A(n)$ is the description of C_n.
on input x, TM simulates $C_{|x|}(x)$

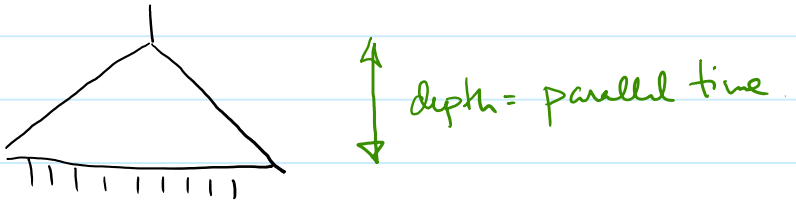
We believe that $NP \not\subseteq P$

Generally believe that $NP \not\subseteq P/poly$.
(i.e. SAT does not have poly-sized circuits, even w/o uniformity constraint)

Many believe that this is the best route to showing $P \neq NP$.

Thursday, April 19, 2018 9:12 PM

Parallelism: uniform circuits allow for a refinement of polynomial time.



NC ("Nick's Class") Hierarchy of logspace uniform circuits.

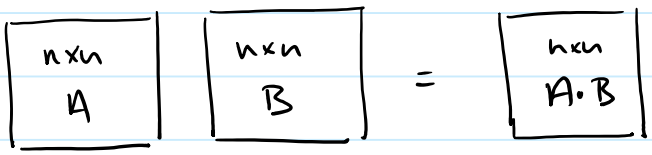
NC_k = languages computable by families of log space uniform circuits with.

- poly # gates.
- depth $O(\log^k n)$

$NC = \bigcup_{k \geq 1} NC_k$ "Efficiently parallelizable problems"

↳ poly log depth.

Example Matrix Multiplication.



What is the parallel complexity of this problem?

work = poly(n).

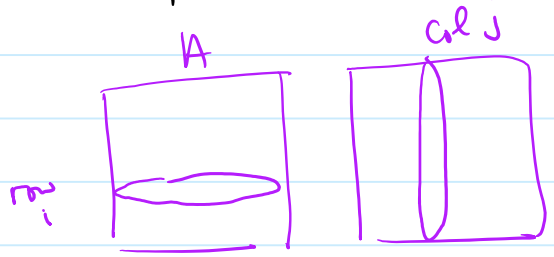
parallel time = $\log^k(n)$ for which k?

Thursday, April 19, 2018 2:04 PM

mult : AND.
add : OR.

Boolean Matrix Multiplication °
matrix entries are 0/1
use Boolean addition & multiplication.
$(1+1) = 1$ "OR".

$$(AB)_{ij} = \bigvee_k (a_{ik} \wedge b_{kj})$$



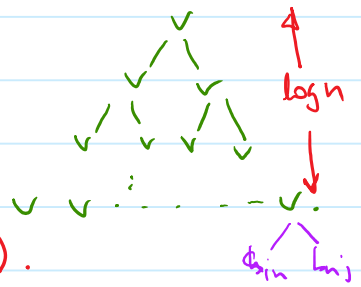
To make output into a single bit $(A, B, i, j) \rightarrow AB_{ij}$

To get all of $A \cdot B$, just have n^2 circuits in parallel.

Boolean Matrix Multiplication $\subseteq NC_1$.

level 1: compute $(a_{ik} \wedge b_{kj})$ $\forall k$ in parallel

Compute n-wise OR with a binary tree of depth $\log n$.



(for single bit output, this is actually a formula).

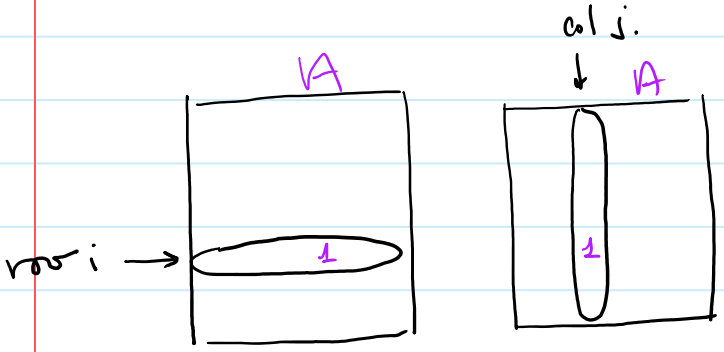
We can use this to show that ST-CONN $\in NC_2$
which implies that $NL \subseteq NC_2$.

Thursday, April 19, 2018 2:04 PM

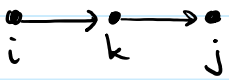
Let A be the adjacency matrix for G w/ self-loops added (1's along the diagonal).

$(A^2)_{ij}$ $\longleftrightarrow$ $\exists$ path from i to j of length ≤ 2

Boolean matrix multiplication.



$\exists$ path of length ≤ 2 from i to j iff $\exists$ k.



$(A^2)_{ij} = 1$ iff $\exists$ k. $A_{ik} \wedge A_{kj}$

The self loops allow for $i=k$ or $k=j$:

A^n represents paths of length $\leq n$. (i.e paths of any length).

$(A^n)_{st} = 1$ iff $\exists$ path from s to t.

Thursday, April 19, 2018 2:04 PM

To compute A^n: $A \to A^2 \to A^4 \cdots \to A^{2^{\log n}}$

$(A^2)^2 \quad A^8 = (A^4)^2$

$\log n$ matrix multiplications.

$\Rightarrow \log^2 n$ depth.

Does P-completeness extend to the class NC?

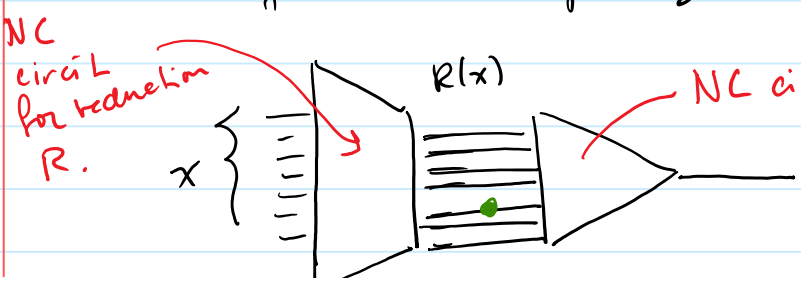
If a P-complete problem is in NC does that imply that NC = P? (Yes).

Need to establish that log-space reductions can be converted to log-space uniform NC circuits.

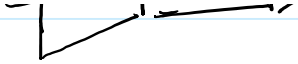
If $L \propto L'$ and $L' \in NC$ then $L \in NC$.

$\propto$ → logspace reductions.

Need to show that an NC circuit can compute the reduction + circuit from a logspace uniform circuit family.



$x \in L \iff R(x) \in L'$.



Thursday, April 19, 2018 2:04 PM

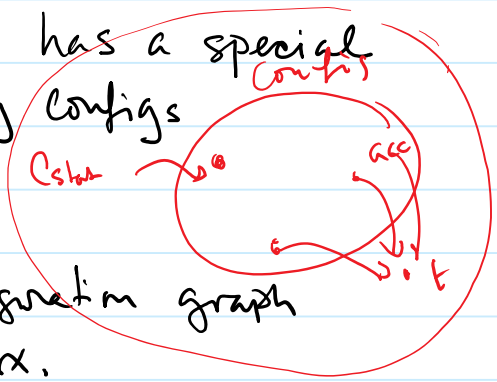
The reduction R takes input x + outputs $R(x)$.

We will only worry about a single bit of $R(x)$
Can run single-bit circuits in parallel to get all of $R(x)$.

Consider a TM M that takes in a pair (x, i) s.t. $i \leq |R(x)|$. It accepts iff the i^{th} bit of $R(x) = 1$. M runs in log space.

The circuit that computes the i^{th} bit of $R(x)$ will compute ST-CONN on the configuration graph for M's computation on x.

Recall that the configuration graph has a special node t to which all accepting configs have an edge.



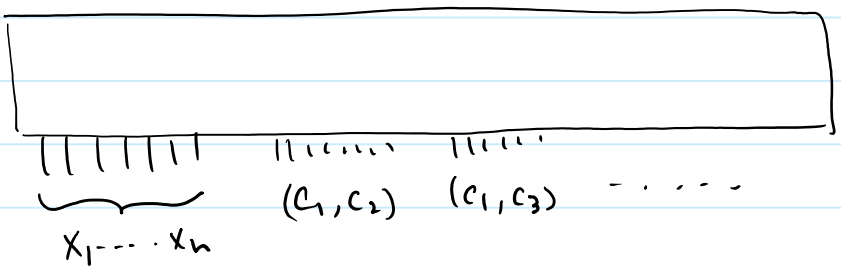
A_x = adjacency matrix for the configuration graph for M's computation on input x.

Output of circuit $\left[(A_x)^n\right]_{start,t}$

A adj matrix for config graph of M on input x

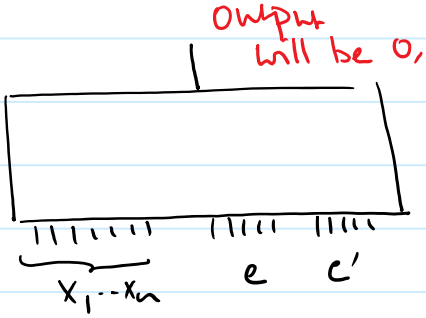
We know how to compute A^n.
Just need to compute A_x.

Friday, April 20, 2018 8:22 AM



← all possible config pairs hard-coded 0/1.

Need to show:



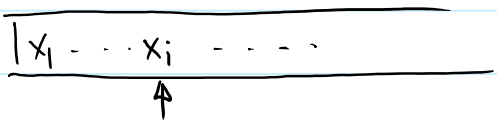
Is (c, c') an edge?

Consider two configurations: output 0, 1, or x_i.

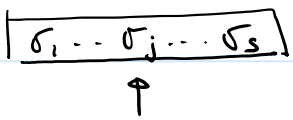
$C = (i, j, \sigma_1 \cdots \sigma_s, q)$ $C' = (i', j', \sigma_1' \cdots \sigma_s', q')$

location of head on the input tape

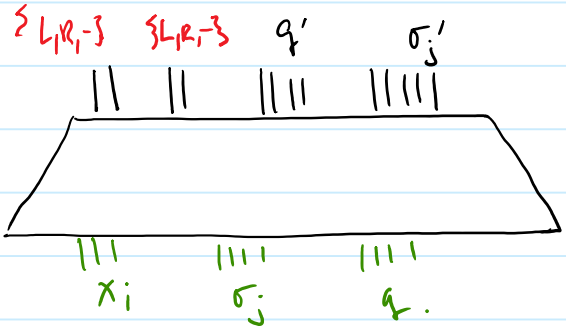
location of head on the work tape.



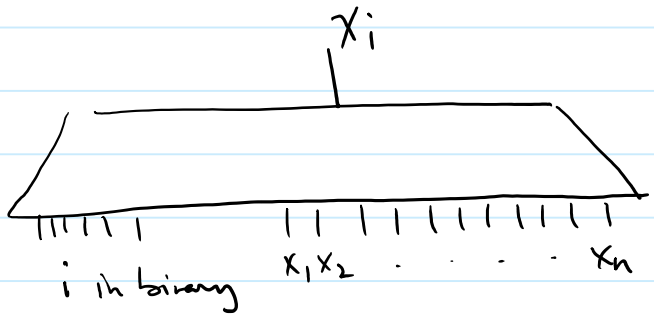
q.



Move from C depends on: x_i σ_j q



Friday, April 20, 2018 8:41 AM



Selection can be done in NC.