Umans Complexity Theory Lectures

Boolean Circuits & NP:

- Uniformity and Advice
- NC hierarchy

1

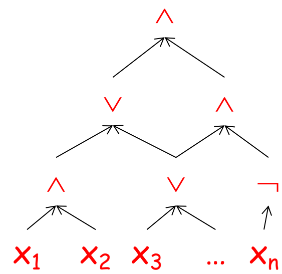
Outline

- Boolean circuits and formulas
- uniformity and advice
- the **NC** hierarchy and parallel computation
- the quest for circuit lower bounds
- a lower bound for formulas

2

Boolean circuits

- **circuit** C
  - directed acyclic graph
  - nodes: AND ($\wedge$); OR ($\vee$); NOT ($\neg$); variables x_i



- C computes function $f:\{0,1\}^n \rightarrow \{0,1\}$.
  - identify C with function f it computes

3

Boolean circuits

- **size** = # gates
- **depth** = longest path from input to output
- **formula (or expression)**: graph is a tree

- Every function $f:\{0,1\}^n \rightarrow \{0,1\}$ is computable by a circuit of size at most $O(n2^n)$
  - AND of n literals for each x such that f(x) = 1
  - OR of up to 2^n such terms

4

Circuit families

- Circuit works only for all inputs of a specific input length n.
- Given function $f: \Sigma^* \rightarrow \{0,1\}$
- Circuit **family** : a circuit for each input length: $C_1, C_2, C_3, \ldots$ = "$\{C_n\}$"
- "$\{C_n\}$ computes f" iff for all x in Σ^*

$$C_{|x|}(x) = f(x)$$

- "$\{C_n\}$ decides L", where L is the language associated with f: For all x in Σ^* ,

  x in L iff $C_{|x|}(x) = f(x) = 1$

5

Connection to TMs

- given TM M running in time T(n) decides language L
- can build circuit family $\{C_n\}$ that decides L
  - size of $C_n = O(T(n)^2)$
  - Proof: CVAL construction used for polytime-completeness proof
- Conclude: $L \in$ **P** implies family of polynomial-size circuits that decides L

6

Uniformity

- Strange aspect of circuit families:
  - can "encode" (potentially uncomputable) information in family specification
- solution: uniformity – require specification is simple to compute

Definition: circuit family $\{C_n\}$ is **logspace uniform** iff there is a TM M that outputs C_n on input 1^n and runs in O(log n) space.

7

Uniformity

Theorem: **P** = languages decidable by logspace uniform, polynomial-size circuit families $\{C_n\}$.

- Proof:
  - already saw ($\Rightarrow$)
  - ($\Leftarrow$) on input x, generate $C_{|x|}$, evaluate it and accept iff output = 1

8

TMs that take advice

- A circuit family $\{C_n\}$ without uniformity constraint is called "non-uniform"
- Regard "non-uniformity" as a limited resource just like time, space, as follows:
  - add read-only "advice" tape to TM M
  - M "decides L with advice A(n)" iff

    $$M(x, A(|x|)) \text{ accepts} \Leftrightarrow x \in L$$
  - note: A(n) depends only on |x|

9

TMs that take advice

- Definition: **TIME(T(n))/f(n)** = the set of those languages L for which:
  - there exists A(n) s.t. $|A(n)| \leq f(n)$
  - TM M decides L with advice A(n) in time T(n)
- most important such class:

$$\mathbf{P/poly} = \cup_k \mathbf{TIME}(\mathbf{n}^k)/\mathbf{n}^k$$

10

TMs that take advice

Theorem: L $\in$ **P/poly** iff L decided by family of (non-uniform) polynomial size circuits.

- Proof:
  - ($\Rightarrow$) C_n from CVAL construction; hardwire advice A(n)
  - ($\Leftarrow$) define A(n) = description of C_n; on input x, TM simulates $C_{|x|}(x)$

11

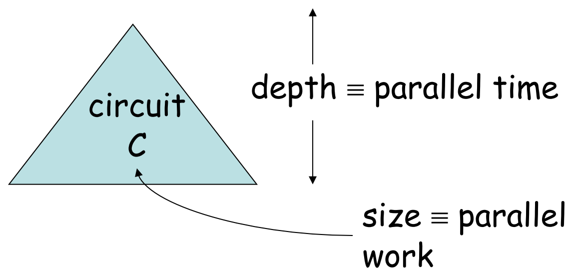
Approach to P/NP

- Believe **NP** $\not\subset$ **P**
  - equivalent: "**NP** does not have uniform, polynomial-size circuits"
- *Even believe* **NP** $\not\subset$ **P/poly**
  - equivalent: "**NP** (or, e.g. SAT) does not have polynomial-size circuits"
  - implies **P** $\neq$ **NP**
  - many believe: best hope for **P** $\neq$ **NP**

12

Parallelism

- uniform circuits allow refinement of polynomial time:



13

Parallelism

- The **NC** ("**N**ick's **C**lass") Hierarchy of logspace uniform circuits:

  NC_k = $O(\log^k n)$ depth, poly(n) size

  $NC = \cup_k NC_k$

- captures "efficiently parallelizable problems"

14

Matrix Multiplication

n x n matrix A | n x n matrix B = n x n matrix AB

- what is the parallel complexity of this problem?
  - work = poly(n)
  - time = log (n)

15

Matrix Multiplication

- two details
  - arithmetic matrix multiplication…

    $A = (a_{i,\,k})$ $B = (b_{k,\,j})$ $(AB)_{i,j} = \Sigma_k\,(a_{i,k} \times b_{k,\,j})$

    … vs. Boolean matrix multiplication:

    $A = (a_{i,\,k})$ $B = (b_{k,\,j})$ $(AB)_{i,j} = \vee_k\,(a_{i,k} \wedge b_{k,\,j})$

  - single output bit: to make matrix multiplication a language: on input A, B, (i, j) output $(AB)_{i,j}$

16

Matrix Multiplication

- Boolean Matrix Multiplication is in $\mathbf{NC_1}$
  - level 1: compute n ANDS: $a_{i,k} \wedge b_{k,j}$
  - next log n levels: tree of ORS

  - n^2 subtrees for all pairs (i, j)
  - select correct one and output

17

Boolean formulas and $\mathbf{NC_1}$

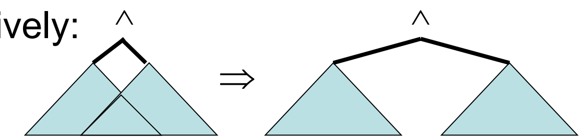
- A formula is a circuit that is a tree with no shared substructures.
- We measure formula size by leaf-size.
- Previous circuit for matrix mult is actually a formula. This is no accident:

Theorem: $L \in \mathbf{NC_1}$ iff decidable by polynomial-size uniform family of Boolean formulas.

18

Boolean formulas and $\mathbf{NC_1}$

- Proof:
  - ($\Rightarrow$) convert a $\mathbf{NC_1}$ circuit **of depth log(n)** into a formula tree
    - recursively:



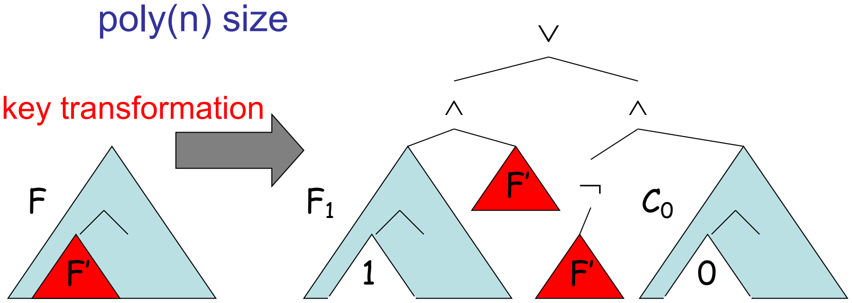
    - note: logspace transformation (stack depth log n, stack record 1 bit – "left" or "right")

Aril 16, 2013

19

Boolean formulas and $\mathbf{NC_1}$

- ($\Leftarrow$) convert formula tree F of size n into formula tree of depth O(log n)
  - note: size $\leq 2^{\text{depth}}$, so new formula has poly(n) size



20

Boolean formulas and $\mathbf{NC_1}$

Let F' be a minimal subtree of formula tree F with size at least n/3

- implies size(F') ≤ 2n/3

– define D(n) = maximum depth required for any size n formula

– Subtrees F_1, F_0, F' all size ≤ 2n/3

$$D(n) \leq D(2n/3) + 3$$

implies depth $D(n) \leq O(\log n)$

21

Relation to other classes

- Clearly **NC** ⊆ **P**
  – recall **P** ≡ uniform poly-size circuits
- $\mathbf{NC_1} \subseteq \mathbf{L}$
  – on input x, compose logspace algorithms for:
    - generating $C_{|x|}$
    - converting to formula
    - FVAL

22

Relation to other classes

- **NL** ⊆ **NC_2**: S-T-CONN ∈ **NC_2**
  - given G = (V, E), vertices s, t
  - A = adjacency matrix (with self-loops)
  - $(A^2)_{i,j} = 1$ iff path of length ≤ 2 from node i to node j
  - $(A^n)_{i,j} = 1$ iff path of length ≤ n from node i to node j
  - compute with depth log n tree of Boolean matrix multiplications, output entry s, t
  - $\log^2 n$ depth total

23

NC vs. P

- Can every efficient algorithm be efficiently parallelized?

$$\mathbf{NC} \stackrel{?}{=} \mathbf{P}$$

- **P**-complete problems least-likely to be parallelizable
  - if **P**-complete problem is in **NC**, then **P** = **NC**
    - Why:

    we use logspace reductions to show problem **P**-complete; **L** in **NC**

24

NC vs. **P**

- Open: Can every uniform, poly-size Boolean circuit family be converted into a uniform, poly-size Boolean formula family?

$$\mathbf{NC_1 \stackrel{?}{=} P}$$

25

NC Hierarchy Collapse

$$NC_1 \subseteq NC_2 \subseteq NC_3 \subseteq NC_4 \subseteq \ldots \subseteq NC$$

<u>Exercise</u>

if $NC_i = NC_{i+1}$, then $NC = NC_i$

(prove for non-uniform versions of classes)

26