CS151
Complexity Theory

Lecture 6

April 19, 2017

Shannon's counting argument

- frustrating fact: *almost all* functions require huge circuits

Theorem (Shannon): With probability at least $1 - o(1)$, a random function

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

requires a circuit of size $\Omega(2^n/n)$.

Shannon's counting argument

- Proof (counting):
  - $B(n) = 2^{2^n}$ = # functions $f:\{0,1\}^n \rightarrow \{0,1\}$
  - # circuits with n inputs + size s, is at most

$$C(n, s) \leq ((n+3)s^2)^s$$

s gates

n+3 gate types

2 inputs per gate

Shannon's counting argument

$$C(n, s) \le ((n+3)s^2)^s$$

- $C(n, c2^n/n) < ((2n)c^2 2^{2n}/n^2)^{(c2^n/n)}$
  $< o(1)2^{2c2^n}$
  $< o(1)2^{2^n}$ (if $c \le ½$)

- probability a random function has a circuit of size $s = (½)2^n/n$ is at most

$$C(n, s)/B(n) < o(1)$$

Shannon's counting argument

- frustrating fact: *almost all* functions require huge **formulas**

Theorem (Shannon): With probability at least $1 - o(1)$, a random function

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

requires a **formula** of size $\Omega(2^n/\log n)$.

Shannon's counting argument

- Proof (counting):
  - $B(n) = 2^{2^n}$ = # functions $f:\{0,1\}^n \rightarrow \{0,1\}$
  - # formulas with n inputs + size s, is at most

$$F(n, s) \leq 4^s 2^s (2n)^s$$

4^s binary trees with s internal nodes

2 gate choices per internal node

2n choices per leaf

Shannon's counting argument

$$F(n, s) \le 4^s 2^s (2n)^s$$

- $F(n, c2^n/\log n) < (16n)^{(c2^n/\log n)}$
  $< 16^{(c2^n/\log n)} 2^{(c2^n)} = (1 + o(1))2^{(c2^n)}$
  $< o(1)2^{2^n}$ (if $c \le ½$)

- probability a random function has a **formula** of size $s = (½)2^n/\log n$ is at most $F(n, s)/B(n) < o(1)$

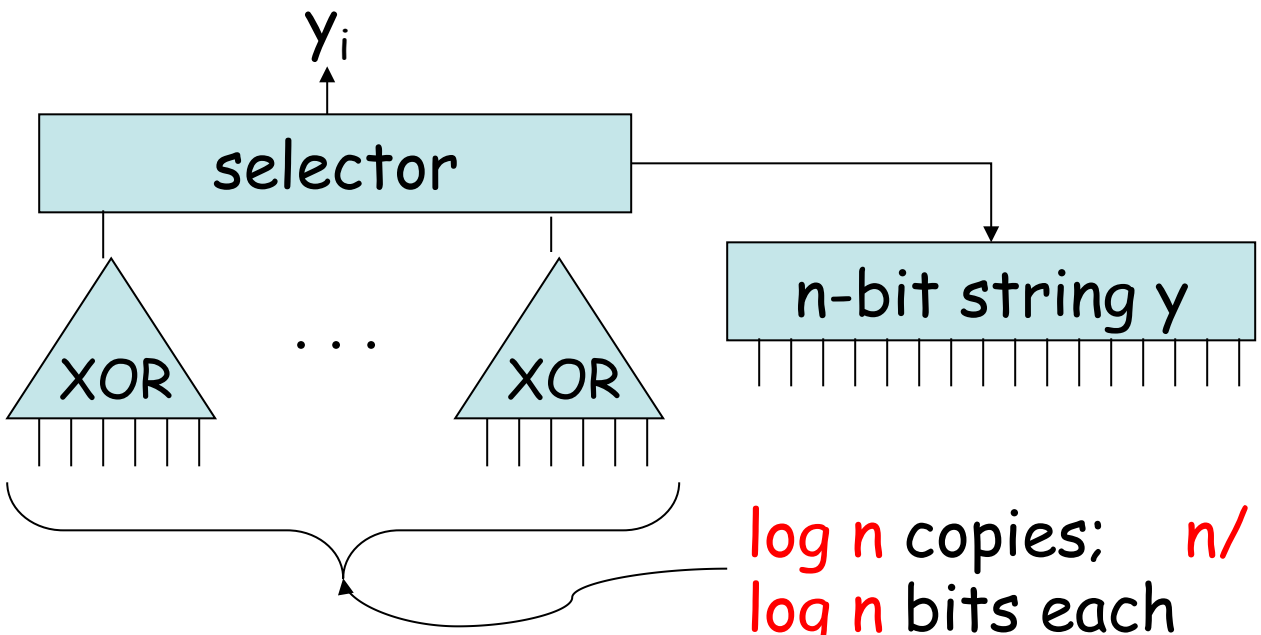
Andreev function

- best formula lower bound for language in NP:

Theorem (Andreev, Hastad '93): the Andreev function requires $(\wedge, \vee, \neg)$-formulas of size at least

$$\Omega(n^{3-o(1)}).$$

Andreev function



the Andreev function $A(x,y)$

$A:\{0,1\}^{2n} \rightarrow \{0,1\}$

Random restrictions

- key idea: given function

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

  restrict by ρ to get f_ρ
  - ρ sets some variables to 0/1, others remain free
- $R(n, \epsilon n)$ = set of restrictions that leave ϵn variables free
- Definition: $L(f)$ = smallest $(\wedge, \vee, \neg)$ formula computing f (measured as leaf-size)

Random restrictions

- observation:

$$E_{\rho \leftarrow R(n, \epsilon n)}[L(f_\rho)] \leq \epsilon L(f)$$

  – each leaf survives with probability ϵ

- may shrink more…

  – propogate constants

Lemma (Hastad 93): for all f

$$E_{\rho \leftarrow R(n, \epsilon n)}[L(f_\rho)] \leq O(\epsilon^{2-o(1)} L(f))$$

Hastad's shrinkage result

- Proof of theorem:
  - Recall: there exists a function

    $$h:\{0,1\}^{\log n} \rightarrow \{0,1\}$$

    for which L(h) > n/2loglog n.
  - hardwire truth table of that function into y to get $A^*(x)$
  - apply random restriction from

    $$R(n, m = 2(\log n)(\ln \log n))$$

    to $A^*(x)$.

The lower bound

- Proof of theorem (continued):
  - probability given XOR is killed by restriction is probability that we "miss it" m times:

$$(1 - (n/\log n)/n)^m \leq (1 - 1/\log n)^m$$

$$\leq (1/e)^{2 \ln \log n} \leq 1/\log^2 n$$

  - probability even one of XORs is killed by restriction is at most:

$$\log n(1/\log^2 n) = 1/\log n < ½.$$

The lower bound

- (1): probability even one of XORs is killed by restriction is at most:

$$\log n(1/\log^2 n) = 1/\log n < ½.$$

- (2): by Markov:

$$\Pr[\, L(A^*_\rho) > 2\, E_{\rho \leftarrow R(n,\, m)}[L(A^*_\rho)] \,] < ½.$$

- Conclude: for *some* restriction ρ
  - all XORs survive, and
  - $L(A^*_\rho) \le 2\, E_{\rho \leftarrow R(n,\, m)}\, [L(A^*_\rho)]$

The lower bound

- Proof of theorem (continued):
  - if all XORs survive, can restrict formula further to compute hard function h
    - may need to add ¬'s

$$L(h) = \mathbf{n}/2\log\log n \le L(A^*_\rho)$$

$$\le 2E_{\rho \leftarrow R(n,\, m)}[L(A^*_\rho)] \le O((m/n)^{2-o(1)} L(A^*))$$

$$\le O\left(((\log n)(\ln \log n)/\mathbf{n})^{\mathbf{2-o(1)}}\, L(A^*) \right)$$

  - implies $\Omega(n^{3-o(1)}) \le L(A^*) \le L(A)$.

Clique

CLIQUE = { (G, k) | G is a graph with a clique of size ≥ k }

(clique = set of vertices every pair of which are connected by an edge)

- CLIQUE is **NP**-complete.

Circuit lower bounds

- We think that **NP** requires exponential-size circuits.
- Where should we look for a problem to attempt to prove this?

- Intuition: “hardest problems” – i.e., **NP-complete problems**

Circuit lower bounds

- Formally:
  - if *any* problem in **NP** requires super-polynomial size circuits
  - then *every* **NP**-complete problem requires super-polynomial size circuits

  - Proof idea: poly-time reductions can be performed by poly-size circuits using a variant of CVAL construction

Monotone problems

- Definition: monotone language = language

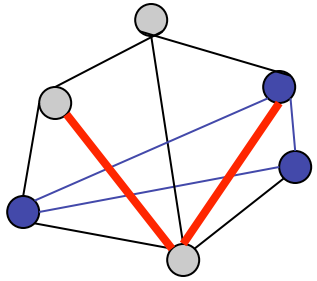
$$L \subset \{0,1\}^*$$

such that $x \in L$ implies $x' \in L$ for all $x \preceq x'$.

  – flipping a bit of the input from 0 to 1 can only change the output from "no" to "yes" (or not at all)

Monotone problems

- some **NP**-complete languages are monotone
  - e.g. CLIQUE (given as adjacency matrix):



  - others: HAMILTON CYCLE, SET COVER…
  - but not SAT, KNAPSACK…

Monotone circuits

A restricted class of circuits:

- Definition: monotone circuit = circuit whose gates are ANDs (∧), ORs (∨), but no NOTs

- can compute exactly the monotone fns.
  - monotone functions closed under AND, OR

Monotone circuits

- A question:

Do all

poly-time computable monotone functions

have

poly-size monotone circuits?

  – recall: true in non-monotone case

Monotone circuits

A monotone circuit for $CLIQUE_{n,k}$

- Input: graph $G = (V,E)$ as adj. matrix, $|V|=n$
  - variable $x_{i,j}$ for each possible edge (i,j)
- $ISCLIQUE(S)$ = monotone circuit that = 1 iff $S \subset V$ is a clique: $\bigwedge_{i,j \in S} x_{i,j}$
- $CLIQUE_{n,k}$ computed by monotone circuit:

$$\bigvee_{S \subset V,\ |S| = k} ISCLIQUE(S)$$

Monotone circuits

- Size of this monotone circuit for $\text{CLIQUE}_{n,k}$:

$$\binom{n}{k}\binom{k}{2}$$

- when $k = n^{1/4}$, size is approximately:

$$\left(\frac{n}{n^{1/4}}\right)^{n^{1/4}}\left(\frac{n^{1/4}}{2}\right)^{2} \approx n^{\Omega\left(n^{1/4}\right)}$$

Monotone circuits

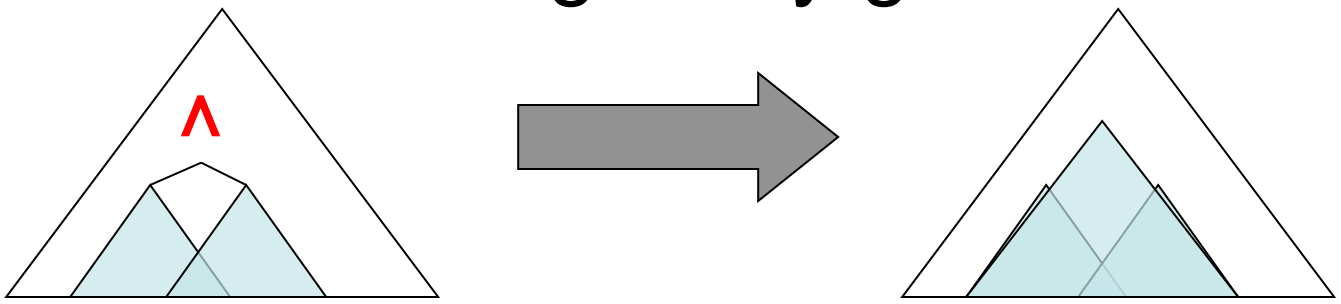
- Theorem (Razborov 85): monotone circuits for $CLIQUE_{n,k}$ with $k = n^{1/4}$ must have size at least

$$2^{\Omega(n^{1/8})}.$$

- Proof:
  - rest of lecture

Proof idea

- "method of approximation"
- suppose C is a monotone circuit for $CLIQUE_{n,k}$
- build another monotone circuit CC that "approximates" C gate-by-gate



Proof idea

- on test collection of positive/negative instances of $\text{CLIQUE}_{n,k}$:
  - local property: few errors at each gate
  - global property: many errors on test collection

- Conclude: C has many gates

Notation

- input: graph G = (V, E)
- variable $x_{j,k}$ for each potential edge (j, k)
- CC(X_1, X_2, … X_m), where $X_i \subset V$, means:

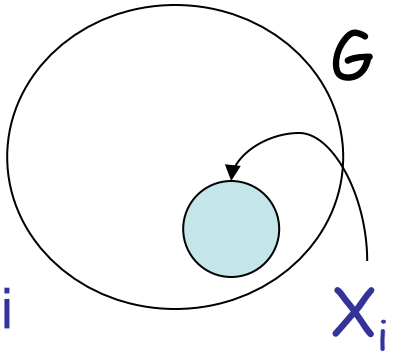
$$\vee_i \left(\wedge_{j,k \in X_i} x_{j,k} \right)$$

- For example: CC(X_1, X_2, … X_m) where the X_i range over all k-subsets of V
  - this is the obvious monotone circuit for $CLIQUE_{n,k}$ from a previous slide.

*[CC( ) = 0; $(\wedge_{j,k \in \emptyset} x_{j,k}) = 1$]

Preview

- approximate circuit CC(X_1, X_2, … X_m)
- n = # nodes
- k = $n^{1/4}$ = size of clique
- h = $n^{1/8}$ = max. size of subsets X_i
  - this is "global property" that ensures lots of errors
  - many graphs G with no k-cliques, but clique on X_i of size h

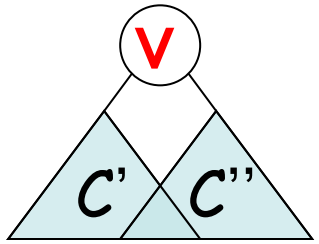


Preview

- approximate circuit CC(X_1, X_2, … X_m)
- $p = n^{1/8} \log n$
- $M = (p - 1)^h h!$
- max # of subsets is M (so $m \leq M$)
  - critical for "local property" that ensures few errors at each gate

Building CC

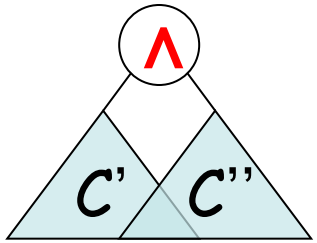
- CC ("crude circuit") for circuit C defined inductively as follows:
  - CC for single variable $x_{j,k}$ is just CC({ j, k })
    - no errors yet!
  - CC for circuit C of form:



  - "approximate OR" of CC for C', CC for C''

Building CC

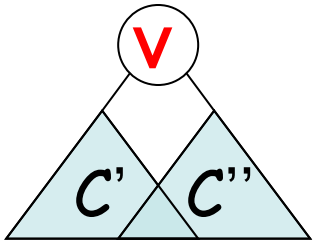
– CC for circuit C of form:



– "approximate AND" of CC for C', CC for C''

– "approximate OR" and "approximate AND" steps introduce errors

Approximate OR



$CC(X_1, X_2, \ldots X_{m'})$ $CC(Y_1, Y_2, \ldots Y_{m''})$

- exact OR:

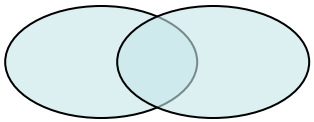
$CC(X_1, X_2, \ldots X_{m'}, Y_1, Y_2, \ldots Y_{m''})$

  - set sizes still ≤ h
  - may be up to 2M sets; need to reduce to M

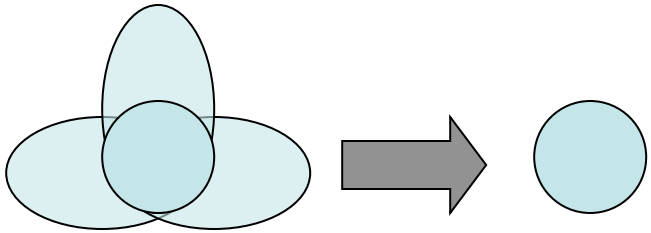
Approximate OR

– throw away sets? bad:many errors

– throw away overlapping sets? – better

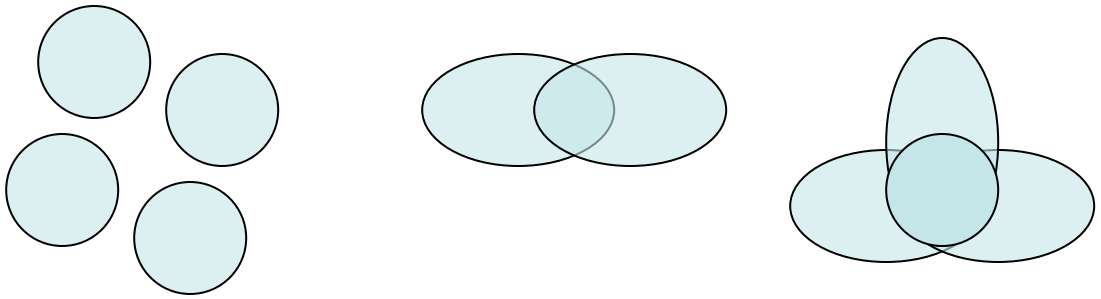


– throw away special configuration of overlapping sets – best



Sunflowers

- Definition: (h, p)-sunflower is a family of p sets ("petals") each of size at most h, such that intersection of every pair is a subset S (the "core").



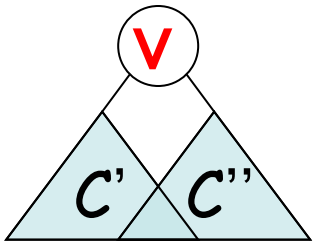
Sunflowers

Lemma (Erdös-Rado): Every family of more than $M = (p-1)^h h!$ sets, each of size at most h, contains an (h, p)-sunflower.

- Proof:
  - not hard
  - in Papadimitriou, elsewhere

Approximate OR

- CC($X_1, X_2, \ldots X_{m'}$)
- CC($Y_1, Y_2, \ldots Y_{m''}$)



- exact OR:

  CC($X_1, X_2, \ldots X_{m'}, Y_1, Y_2, \ldots Y_{m''}$)

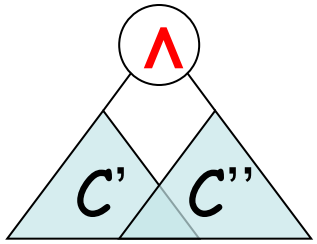
  – while more than M sets, find (h, p)-sunflower; replace with its core ("**pluck**")

- approximate OR:

  CC(**pluck**($X_1, X_2, \ldots X_{m'}, Y_1, Y_2, \ldots Y_{m''}$))

Approximate AND

- CC($X_1, X_2, \ldots X_{m'}$)
- CC($Y_1, Y_2, \ldots Y_{m''}$)
- (close to) exact AND:



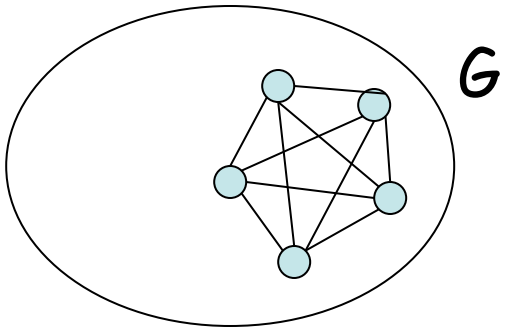
$$CC(\{(X_i \cup Y_j) : 1 \le i \le m', 1 \le j \le m''\})$$

  - some sets may be larger than h; discard them
  - may be up to M^2 sets. While > M sets, find (h, p)-sunflower; replace with its core ("**pluck**")
- approximate AND:

$$CC(\textbf{pluck} (\{(X_i \cup Y_j) : |X_i \cup Y_j| \le h \}))$$

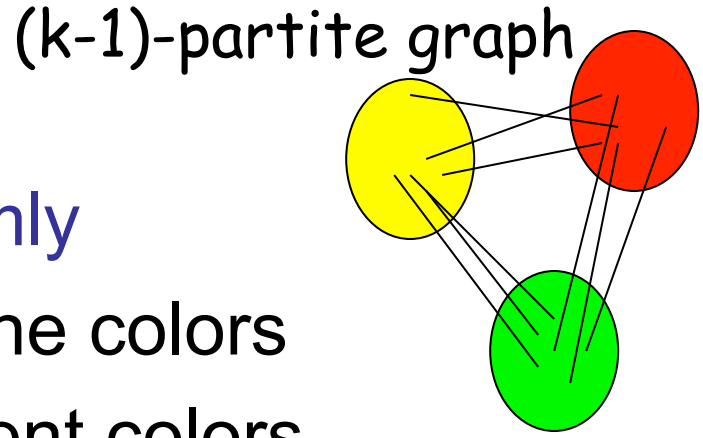
Test collection

- Positive instances: all graphs G on n nodes with a k-clique and no other edges.



Test collection

- Negative instances:
  - k-1 colors
  - color each node uniformly at random with one of the colors
  - edge (x, y) iff x, y different colors
  - no k-clique
  - include graphs in their multiplicities
    - makes analysis easier

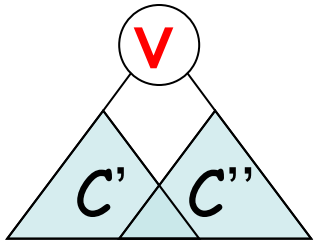


"Local" analysis

- "false positive":
  - negative example
  - gate is supposed to output 0, but our CC outputs 1

Lemma: each approximation step introduces at most $M^2(k-1)^n/2^p$ false positives.

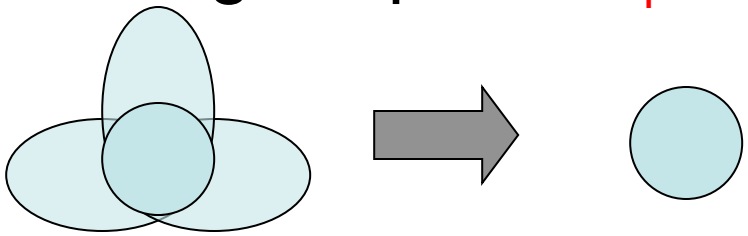
“Local” analysis



- Proof:
  - case 1: OR

    $CC(X_1, X_2, \ldots X_{m'})$ $CC(Y_1, Y_2, \ldots Y_{m''})$

    $CC(\textbf{pluck}(X_1, X_2, \ldots X_{m'}, Y_1, Y_2, \ldots Y_{m''}))$
  - given “plucking”: replace $Z_1 \ldots Z_p$ with Z



  - bad case: clique on Z, and each petal is missing at least one edge

"Local" analysis

– what is the probability of a repeated color in each Z_i but no repeated colors in Z?

$$\Pr[R(Z_1) \wedge R(Z_2) \ldots R(Z_p) \wedge \neg R(Z)]$$

$$\leq \Pr[R(Z_1) \wedge R(Z_2) \ldots R(Z_p) | \neg R(Z)]$$

(definition of conditional probability)

$$= \prod_i \Pr[R(Z_i) \mid \neg R(Z)]$$

(independent events given no repeats in Z)

$$\leq \prod_i \Pr[R(Z_i)]$$

(obviously larger)

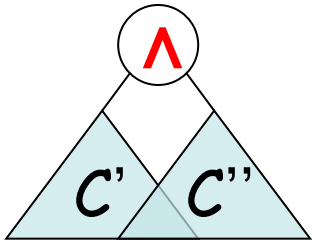
event R(S) = repeated colors in S

"Local" analysis

- for every pair of vertices in Z_i, probability of same color is $1/(k-1)$
- $R(Z_i) \leq (h \text{ choose } 2)/(k-1) \leq ½$
- $\prod_i \Pr[R(Z_i)] \leq (½)^p$
- # negative examples is $(k-1)^n$
- # false positives in given plucking step is at most $(½)^p(k-1)^n$
- at most M plucking steps
- # false positives at OR $\leq M(½)^p(k-1)^n$

“Local” analysis

– case 2: AND



$CC(X_1, X_2, \ldots X_{m'})$ $CC(Y_1, Y_2, \ldots Y_{m''})$

$CC(\textbf{pluck}(\{(X_i \cup Y_j) : |X_i \cup Y_j| \leq h \}))$

– discarding sets $(X_i \cup Y_j)$ larger than h can only make circuit accept fewer examples

- no false positives here

“Local” analysis

– up to M^2 pluckings

– each introduces at most

$$(½)^p(k\text{-}1)^n$$

false positives (previous slides)

– # false positives at AND ≤ $M^2(½)^p(k\text{-}1)^n$

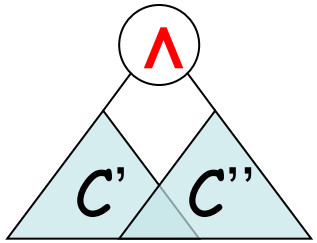
"Local" analysis

- "false negative":
  - positive example;
  - gate is supposed to output 1, but our CC outputs 0

Lemma: each approximation step introduces at most $M^2 \binom{n-h-1}{k-h-1}$ false negatives.

“Local” analysis

- Proof:
  - Case 1: OR
  - plucking can only make circuit accept more examples
    - no false negatives here.



  - Case 2: AND

$$CC(X_1, X_2, \ldots X_{m'}) \qquad CC(Y_1, Y_2, \ldots Y_{m''})$$

$$CC(\textbf{pluck}(\{(X_i \cup Y_j) : |X_i \cup Y_j| \leq h\}))$$

    - for positive examples: clique on X_i and clique on $Y_j \Rightarrow$ clique on $X_i \cup Y_j$ (no false negatives until discard $X_i \cup Y_j$ sets)

"Local" analysis

- discarding set $Z = (X_i \cup Y_j)$ larger than h may introduce false negatives
- any clique that includes Z is a problem; there are at most

$$\binom{n-|Z|}{k-|Z|} \leq \binom{n-h-1}{k-h-1}$$

  such positive examples, since $|Z|>h$
- at most M^2 such deletions
- we've seen plucking doesn't matter

"Global" analysis

Lemma: every non-trivial CC outputs 1 on at least ½ of the negative examples.

- Proof:
  - CC contains some set X of size at most h
  - accepts all neg. examples with different colors in X
  - probability X has repeated colors is

$$R(X) \leq (h \text{ choose } 2)/(k-1) \leq ½$$

  - probability over negative examples that CC accepts is at least ½.

Finishing up

- First possibility: trivial CC, rejects all positive examples
  - every positive example must have been false negative at some gate
  - number of gates must be at least:

$$\binom{n}{k} / M^2 \binom{n-h-1}{k-h-1}$$

Finishing up

- Second possibility: CC accepts at least ½ of negative examples
  - every negative example must have been false positive at some gate
  - number of gates must be at least:

$$\frac{1}{2}(k-1)^n / M^2 2^{-p}(k-1)^n$$

Finishing up

$$\binom{n}{k} / M^2 \binom{n-h-1}{k-h-1}$$

$$\frac{1}{2}(k-1)^n / M^2 2^{-p}(k-1)^n$$

Both quantities are at least $2^{\Omega(n^{1/8})}$

Conclusions

- A question (true in non-monotone case):

  Do all

  poly-time computable monotone functions

  have

  poly-size monotone circuits?

- if yes, then we would have just proved **P ≠ NP**
  - why?

Conclusions

- unfortunately, answer is no

- Razborov later showed similar (super-polynomial) lower bound for MATCHING, which is in **P**…