Natural Proofs

- Razborov and Rudich defined the following "natural" format for circuit lower bounds:
  - identify property $P$ of functions $f : \{0,1\}^* \rightarrow \{0,1\}$
  - $P = \cup_n P_n$ is a natural property if:
    - (useful) $\forall n f_n \in P_n$ implies $f$ does not have poly-size circuits
      \[ f_n \in P_n \implies \text{ckt size } \geq s(n) > \text{poly}(n) \]
    - (constructive) can decide $f_n \in P_n$ in poly time given the truth table of $f_n$
    - (large) at least $(\frac{1}{2})^{O(n)}$ fraction of all $2^n$ functions on $n$ bits are in $P_n$
  - show some function family $g = \{g_n\}$ is in $P_n$

- all known circuit lower bound techniques are natural for a suitably parameterized version of the definition

**Theorem** (RR): if there is a $2^n$-OWF, then there is no natural property $P_n$.
  - factoring believed to be $2^n$-OWF
  - general version also rules out natural properties useful for proving many other separations, under similar cryptographic assumptions

Proof:

- main idea: natural property $P_n$ can efficiently distinguish pseudorandom functions from truly random functions
  - but cryptographic assumption implies existence of pseudorandom functions for which this is impossible

Proof (continued)

- Recall: assuming One-Way-Permutations $f_k : \{0,1\}^k \rightarrow \{0,1\}^k$ that are not invertible by $2^k$ size circuits
  - we constructed PRG $G : \{0,1\}^k \rightarrow \{0,1\}^k \times \{0,1\}^k$ $G(x) = (y_1, y_2)$
  - no circuit $C$ of size $s = 2^k$ for which $|\Pr_x[C(G(x)) = 1] - \Pr_z[C(z) = 1]| > 1/s$
    \[ (\text{BMY construction with slightly modified parameters}) \]

Graphically:

\[
\begin{array}{c}
\text{x} \\
\text{y}_1 \\
\text{y}_2
\end{array}
\]
Proof (continued)

• A function $F: \{0,1\}^k \rightarrow \{0,1\}^{2^n}$
  (set $n = k$)

  Given $x$, i, can compute $i$-th output bit
  in time $n \cdot \text{poly}(k)$

  height $n - \log k$

  each $x$, defines a poly-time computable function $f_x$

$\bullet$ $f_x$ in poly-time $\Rightarrow$ for all $x$: $f_x \notin \text{P}_n$
  (useful)

$\bullet$ $\Pr_{g \in \text{P}_n}[g = 1] \geq (1/2)^{O(n)}$ (large)

$\bullet$ constructive: exists circuit $T: \{0,1\}^{2^n} \rightarrow \{0,1\}$
  of size $2^{O(n)}$ for which

  $|\Pr_x[T(f_x) = 1] - \Pr_g[T(g) = 1]| \geq (1/2)^{O(n)}$

  $\forall f \in \text{P}_n \Rightarrow f$ does not have poly-size circuits
  (constructive)

$\bullet$ if $f \in \text{P}_n$?

  in poly time given truth table

  of $f$ at least $(1/2)^{O(n)}$ fraction of all $2^{2^n}$ fn.s. on $n$-bits
  in $\text{P}_n$. (large)
Proof (continued)

• $|\Pr_x[T(f_x) = 1] - \Pr_y[T(g) = 1]| \geq (1/2)^O(n)$

distribution $D_4$: pick roots of red subtrees independently from $(0,1)^k$

June 1, 2023
CS151 Lecture 18

13

Proof (continued)

• $|\Pr_x[T(f_x) = 1] - \Pr_y[T(g) = 1]| \geq (1/2)^O(n)$

distribution $D_5$: pick roots of red subtrees independently from $(0,1)^k$

June 1, 2023
CS151 Lecture 18

14

Proof (continued)

• $|\Pr_x[T(f_x) = 1] - \Pr_y[T(g) = 1]| \geq (1/2)^O(n)$

distribution $D_6$: pick roots of red subtrees independently from $(0,1)^k$

June 1, 2023
CS151 Lecture 18

15

Proof (continued)

• $|\Pr_x[T(f_x) = 1] - \Pr_y[T(g) = 1]| \geq (1/2)^O(n)$

distribution $D_7$: pick roots of red subtrees independently from $(0,1)^k$

June 1, 2023
CS151 Lecture 18

16

Proof (continued)

• $|\Pr_x[T(f_x) = 1] - \Pr_y[T(g) = 1]| \geq (1/2)^O(n)$

distribution $D_{5,1}$: pick roots of red subtrees independently from $(0,1)^k$

June 1, 2023
CS151 Lecture 18

17

Proof (continued)

• $|\Pr_x[T(f_x) = 1] - \Pr_y[T(g) = 1]| \geq (1/2)^O(n)$

June 1, 2023
CS151 Lecture 18

18

– For some $i$: $|\Pr_x[T(D_i) = 1] - \Pr_y[T(D_i) = 1]| \geq (1/2)^{n/2} = (1/2)^{O(n)}$
Proof (continued)

For some $i$:

$|\Pr[T(D_i) = 1] - \Pr[T(D_{i-1}) = 1]| \geq (1/2)^{O(n)/2} = (1/2)^{O(n)}$

fix values at roots of all other subtrees to preserve difference

$D_i$: distribution $D_i$ after fixing

$D_{i-1}$: distribution $D_i$ after fixing

Natural Proofs

To prove circuit lower bounds, we must either:

- Violate largeness: seize upon an incredibly specific feature of hard functions (one not possessed by a random function ! )

- Violate constructivity: identify a feature of hard functions that cannot be computed efficiently from the truth table

- No "non-natural property" known for all but the very weakest models...
"We do not conclude that researchers should give up on proving serious lower bounds. Quite the contrary, by classifying a large number of techniques that are unable to do the job, we hope to focus research in a more fruitful direction. Pessimism will only be warranted if a long period of time passes without the discovery of a non-naturalizing lower bound proof."

Rudich and Razborov 1994

Moral

• To resolve central questions:
  – avoid relativizing arguments
    • use PCP theorem and related results
    • focus on circuits, etc...
  – avoid constructive arguments
  – avoid arguments that yield lower bounds for random functions

Course Summary

• Time and space
  – hierarchy theorems
  – FVAL in L
  – CVAL P-complete
  – QSAT PSPACE-complete
  – succinct CVAL EXP-complete
Course summary

• Non-determinism
  – NTIME hierarchy theorem
  – "NP-intermediate" problems (Ladner’s Theorem)
  – unary languages (likely) not NP-complete
  – Savitch’s Theorem
  – Immerman-Szelepcsényi Theorem

Problem sets:
  – sparse languages (likely) not NP-complete

– formula lower bound (Andreev, Hastad)
– monotone circuit lower bound (Razborov)

Problem sets:
  – Barrington’s Theorem
  – formula lower bound for parity

Course summary

• Randomness
  – polynomial identity testing + Schwartz-Zippel
  – unique-SAT (Valiant-Vazirani Theorem)
  – Blum-Micali-Yao PRG
  – Nisan-Wigderson PRG
  – worst-case hardness ⇒ average-case hardness
  – Trevisan extractor

Problem sets:
  – Goldreich-Levin hard bit

Course summary

• Alternation
  – QSAT, complete for levels of the PH
  – Karp-Lipton theorem
  – BPP in PH

Problem sets:
  – approximate counting + sampling with an NP-oracle
  – VC-dimension is \( \Sigma_2 \)-complete
  – the class \( S_2^P \) (final)

Course summary

• Counting
  – #matching is \#P-complete

Problem sets:
  – permanent is \#P-complete
  – Toda’s theorem: \( \text{PH} \subseteq \text{P}^{\#P} \)

Course summary

• Interaction
  – IP = PSPACE
  – GI in \( \text{NP} \cap \text{coAM} \)
  – using NW PRG for MA, variant for AM
  – hardness of approximation, PCPs
  – elements of the PCP theorem

Problem sets:
  – BLR linearity test
  – Clique hard to approximate to within \( N^\epsilon \)
Course summary

- **Barriers to progress**
  - oracles rule out relativizing proofs
  - "natural proofs" rule out many circuit lower bound techniques

Course summary

- Time and space: L, P, PSPACE, EXP
- Non-determinism: NL, NP, coNP, NEXP
- Non-uniformity: NC, P/poly
- Randomness: RL, ZPP, RP, coRP, BPP
- Alternation: PH, PSPACE
- Counting: #P
- Interaction: IP, MA, AM, PCP[log n, 1]

The big picture

- All classes on previous slide are probably distinct, except:
  - P, ZPP, RP, coRP, BPP (probably all equal)
  - L, RL (probably all equal; NL?)
  - NP, MA, AM (probably all equal)
  - IP = PSPACE
  - PCP[log n, 1] = NP
- Only real separations we know separate classes delimiting same resource:
  - e.g. L ≠ PSPACE, NP ≠ NEXP

The big picture

Remember:

possible explanation for failure to prove conjectured separations…

…is that they are false

The big picture

- Important techniques/ideas:
  - simulation and diagonalization
  - reductions and completeness
  - self-reducibility
  - encoding information using low-degree polynomials
  - randomness
  - others…

The big picture

- I hope you take away:
  - an ability to extract the essential features of a problem that make it hard/easy…
  - knowledge and tools to connect computational problems you encounter with larger questions in complexity
  - background needed to understand current research in this area
The big picture

- background to contribute to current research in this area
  - many open problems
  - young field
  - try your hand…