Natural Proofs

Lecture 25

Weak techniques are indeed weak!

Circuit Lower-Bounds

Circuit Lower-Bounds

- To prove that a (boolean) function family f has no small circuit family

Circuit Lower-Bounds

- To prove that a (boolean) function family f has no small circuit family
  - i.e., non-uniform lower-bound

Circuit Lower-Bounds

- To prove that a (boolean) function family f has no small circuit family
  - i.e., non-uniform lower-bound
- What does a proof look like (often)?

Circuit Lower-Bounds

- To prove that a (boolean) function family f has no small circuit family
  - i.e., non-uniform lower-bound
- What does a proof look like (often)?
  - Some (more general) property Φ that f has

Circuit Lower-Bounds

- To prove that a (boolean) function family f has no small circuit family
  - i.e., non-uniform lower-bound
- What does a proof look like (often)?
  - Some (more general) property Φ that f has
  - Show that functions with Φ have no small circuits

Circuit Lower-Bounds

- To prove that a (boolean) function family f has no small circuit family
  - i.e., non-uniform lower-bound
- What does a proof look like (often)?
  - Some (more general) property Φ that f has
  - Show that functions with Φ have no small circuits
    - Being able to show that for Φ might require it to be a nice (natural) property

Features of a "Natural Property" - 1

Features of a "Natural Property" - 1

- For each length n, Φ holds for a "significant" fraction of all functions on $\{0,1\}^n$

Features of a "Natural Property" - 1

- For each length n, Φ holds for a "significant" fraction of all functions on $\{0,1\}^n$
  - Writing functions on $\{0,1\}^n$ as an N-bit string ($N=2^n$), there are 2^N such functions

Features of a “Natural Property” - 1

- For each length n, Φ holds for a “significant” fraction of all functions on $\{0,1\}^n$
  - Writing functions on $\{0,1\}^n$ as an N-bit string ($N=2^n$), there are 2^N such functions
  - Require at least $1/N$ fraction to have Φ

Motivation

Motivation

- Often $\Phi(f)$ just says if complexity(f) > threshold, according to some complexity measure and for some threshold

Motivation

- Often $\Phi(f)$ just says if complexity(f) > threshold, according to some complexity measure and for some threshold
- Formal complexity measure: $m(\text{literal}) \leq 1$; $m(f \wedge g)$, $m(f \vee g) \leq m(f)+m(g)$

Motivation

- Often $\Phi(f)$ just says if complexity(f) > threshold, according to some complexity measure and for some threshold
- Formal complexity measure: $m(\text{literal}) \leq 1$; $m(f \wedge g)$, $m(f \vee g) \leq m(f)+m(g)$
  - e.g. $m(f) := 1 + FC(f)$, where $FC(f)$ is formula complexity of f

Motivation

- Often $\Phi(f)$ just says if complexity(f) > threshold, according to some complexity measure and for some threshold
- **Formal complexity measure**: $m(\text{literal}) \le 1$; $m(f \wedge g)$, $m(f \vee g) \le m(f)+m(g)$
  - e.g. $m(f) := 1 + FC(f)$, where $FC(f)$ is formula complexity of f
  - In fact, for any m, $m(f) \le 1 + FC(f)$

Motivation

- Often $\Phi(f)$ just says if complexity(f) > threshold, according to some complexity measure and for some threshold
- Formal complexity measure: m(literal) $\leq$ 1; m($f \wedge g$), m($f \vee g$) $\leq$ m(f)+m(g)
  - e.g. m(f) := 1 + FC(f), where FC(f) is formula complexity of f
  - In fact, for any m, m(f) $\leq$ 1 + FC(f)
- Such an m does not single out a few functions for high complexity

Motivation

- Often $\Phi(f)$ just says if complexity(f) > threshold, according to some complexity measure and for some threshold
- **Formal complexity measure**: $m(\text{literal}) \leq 1$; $m(f \wedge g), m(f \vee g) \leq m(f)+m(g)$
  - e.g. $m(f) := 1 + FC(f)$, where $FC(f)$ is formula complexity of f
  - In fact, for any m, $m(f) \leq 1 + FC(f)$
- Such an m does not single out a few functions for high complexity
  - If $m(f_n) > c$ for any f_n, then for 1/4th functions f'_n, $m(f'_n) > c/4$

Motivation

- Often $\Phi(f)$ just says if complexity(f) > threshold, according to some complexity measure and for some threshold
- Formal complexity measure: $m(\text{literal}) \leq 1$; $m(f \wedge g)$, $m(f \vee g) \leq m(f)+m(g)$
  - e.g. $m(f) := 1 + FC(f)$, where $FC(f)$ is formula complexity of f
  - In fact, for any m, $m(f) \leq 1 + FC(f)$
- Such an m does not single out a few functions for high complexity
  - If $m(f_n) > c$ for any f_n, then for 1/4th functions f'_n, $m(f'_n) > c/4$
    - $f = g \oplus h = (g \wedge \neg h) \vee (\neg g \wedge h)$. i.e., partition into tuples $(g, \neg g, h, \neg h)$ such that at least one of them must be complex.

Features of a "Natural Property" - 2

Features of a "Natural Property" - 2

- Φ can be "efficiently" checked given the truth-table

Features of a "Natural Property" - 2

- Φ can be "efficiently" checked given the truth-table
  - $\Phi(f_n)$ can be evaluated in time poly(N)

Features of a "Natural Property" - 2

- Φ can be "efficiently" checked given the truth-table
  - $\Phi(f_n)$ can be evaluated in time poly(N)
- Motivation?

Features of a "Natural Property" - 2

- Φ can be "efficiently" checked given the truth-table
  - $\Phi(f_n)$ can be evaluated in time poly(N)
- Motivation?
  - Do not have examples of effectively using very complex properties

Features of a "Natural Property" - 2

- Φ can be "efficiently" checked given the truth-table
  - $\Phi(f_n)$ can be evaluated in time poly(N)
- Motivation?
  - Do not have examples of effectively using very complex properties
  - Opportunity?

Natural Proof

Natural Proof

- Natural proof that g_n has no low-complexity (small/shallow) circuit family will

Natural Proof

- Natural proof that g_n has no low-complexity (small/shallow) circuit family will
  - Define (implicitly or explicitly) a natural property Φ

Natural Proof

- Natural proof that g_n has no low-complexity (small/shallow) circuit family will
  - Define (implicitly or explicitly) a natural property Φ
    - Φ holds for > 1/N of functions on $\{0,1\}^n$ ($N=2^n$)

Natural Proof

- Natural proof that g_n has no low-complexity (small/shallow) circuit family will
  - Define (implicitly or explicitly) a natural property Φ
    - Φ holds for > $1/N$ of functions on $\{0,1\}^n$ ($N=2^n$)
    - $\Phi(f_n)$ can be evaluated in time poly(N)

Natural Proof

- Natural proof that g_n has no low-complexity (small/shallow) circuit family will
  - Define (implicitly or explicitly) a natural property Φ
    - Φ holds for > $1/N$ of functions on $\{0,1\}^n$ ($N=2^n$)
    - $\Phi(f_n)$ can be evaluated in time poly(N)
  - Show that $\Phi(g_n) = 1$

Natural Proof

- Natural proof that g_n has no low-complexity (small/shallow) circuit family will
  - Define (implicitly or explicitly) a natural property Φ
    - Φ holds for > 1/N of functions on $\{0,1\}^n$ ($N=2^n$)
    - $\Phi(f_n)$ can be evaluated in time poly(N)
  - Show that $\Phi(g_n) = 1$
  - Show that if f_n has a low-complexity circuit family, then $\Phi(f_n)=0$ (i.e., Φ avoids functions of low circuit-complexity)

Natural Proof: Example

- PARITY doesn't have depth d AC circuits
- Define a natural property Φ
  - Φ holds for 1/N of functions on $\{0,1\}^n$
  - $\Phi(f_n)$ can be evaluated in time poly(N)
- Show that Φ(PARITY)=1
- Show that if f_n has a low-complexity circuit family, then $\Phi(f_n)=0$ (i.e., Φ avoids functions of low circuit-complexity)

Natural Proof: Example

- PARITY doesn't have depth d AC circuits
- Define a natural property Φ
  - Φ holds for 1/N of functions on $\{0,1\}^n$
  - $\Phi(f_n)$ can be evaluated in time poly(N)
- Show that Φ(PARITY)=1
- Show that if f_n has a low-complexity circuit family, then $\Phi(f_n)=0$ (i.e., Φ avoids functions of low circuit-complexity)

Φ: Not constant after restricting to n^ϵ vars

Natural Proof: Example

- PARITY doesn't have depth d AC circuits
- Define a natural property Φ
  - Φ holds for 1/N of functions on $\{0,1\}^n$
  - $\Phi(f_n)$ can be evaluated in time poly(N)
- Show that Φ(PARITY)=1
- Show that if f_n has a low-complexity circuit family, then $\Phi(f_n)=0$ (i.e., Φ avoids functions of low circuit-complexity)

Φ: Not constant after restricting to n^ϵ vars

Exercise

Natural Proof: Example

- PARITY doesn't have depth d AC circuits
- Define a natural property Φ
  - Φ holds for 1/N of functions on $\{0,1\}^n$
  - $\Phi(f_n)$ can be evaluated in time poly(N)
- Show that Φ(PARITY)=1
- Show that if f_n has a low-complexity circuit family, then $\Phi(f_n)=0$ (i.e., Φ avoids functions of low circuit-complexity)

Φ: Not constant after restricting to n^ϵ vars

Exercise

Brute-force in N^2 time

Natural Proof: Example

- PARITY doesn't have depth d AC circuits
- Define a natural property Φ
  - Φ holds for 1/N of functions on $\{0,1\}^n$
  - $\Phi(f_n)$ can be evaluated in time poly(N)
- Show that Φ(PARITY)=1
- Show that if f_n has a low-complexity circuit family, then $\Phi(f_n)=0$ (i.e., Φ avoids functions of low circuit-complexity)

Φ: Not constant after restricting to n^ϵ vars

Exercise

Brute-force in N^2 time

Not const. till last var

Natural Proof: Example

- PARITY doesn't have depth d AC circuits
- Define a natural property Φ
  - Φ holds for 1/N of functions on $\{0,1\}^n$
  - $\Phi(f_n)$ can be evaluated in time poly(N)
- Show that $\Phi(\text{PARITY})=1$
- Show that if f_n has a low-complexity circuit family, then $\Phi(f_n)=0$ (i.e., Φ avoids functions of low circuit-complexity)

Φ: Not constant after restricting to n^{ϵ} vars

Exercise

Brute-force in N^2 time

Not const. till last var

Switching Lemma: Depth d AC circuit becomes depth 2, restricted to n^{δ} vars. Can fix to 0 or 1 by restricting $n^{\delta}/2$ more vars.

Limitations of Natural Proofs

Limitations of Natural Proofs

- We don't expect natural proofs to show that any function family is not in P/poly

Limitations of Natural Proofs

- We don't expect natural proofs to show that any function family is not in P/poly
  - Natural properties cannot avoid all functions in P/poly

Limitations of Natural Proofs

- We don't expect natural proofs to show that any function family is not in P/poly
  - Natural properties cannot avoid all functions in P/poly
  - Unless some widely-believed assumptions in cryptography are false!

Limitations of Natural Proofs

- We don't expect natural proofs to show that any function family is not in P/poly
  - Natural properties cannot avoid all functions in P/poly
  - Unless some widely-believed assumptions in cryptography are false!
- Note that we know (non-constructively) that there are function families which need exponential-sized circuit families

Limitations of Natural Proofs

- We don't expect natural proofs to show that any function family is not in P/poly
  - Natural properties cannot avoid all functions in P/poly
  - Unless some widely-believed assumptions in cryptography are false!
- Note that we know (non-constructively) that there are function families which need exponential-sized circuit families
  - Not a natural proof: property Φ involved (whether f_n has a small circuit) is not efficient to evaluate

Limitations of Natural Proofs

- We don't expect natural proofs to show that any function family is not in P/poly
  - Natural properties cannot avoid all functions in P/poly
  - Unless some widely-believed assumptions in cryptography are false!
- Note that we know (non-constructively) that there are function families which need exponential-sized circuit families
  - Not a natural proof: property Φ involved (whether f_n has a small circuit) is not efficient to evaluate
  - But doesn't give an "explicit" function (say NP function)

Limitations of Natural Proofs

Limitations of Natural Proofs

- If (very strong) one-way functions exist, natural properties cannot avoid all P/poly functions

Limitations of Natural Proofs

- If (very strong) one-way functions exist, natural properties cannot avoid all P/poly functions
  - If (very strong) one-way functions exist can create pseudorandom functions

Limitations of Natural Proofs

- If (very strong) one-way functions exist, natural properties cannot avoid all P/poly functions
  - If (very strong) one-way functions exist can create pseudorandom functions
    - A distribution of efficient (P/poly) functions

Limitations of Natural Proofs

- If (very strong) one-way functions exist, natural properties cannot avoid all P/poly functions
  - If (very strong) one-way functions exist can create pseudorandom functions
    - A distribution of efficient (P/poly) functions
    - Indistinguishable from random functions

Limitations of Natural Proofs

- If (very strong) one-way functions exist, natural properties cannot avoid all P/poly functions
  - If (very strong) one-way functions exist can create pseudorandom functions
    - A distribution of efficient (P/poly) functions
    - Indistinguishable from random functions
  - But a natural property that avoids P/poly can be used to distinguish any distribution of P/poly functions from random functions

Pseudorandom Functions

Pseudorandom Functions

- Pseudorandom function:

Pseudorandom Functions

- Pseudorandom function:
  - A small ($2^{\text{polylog}(N)}$ sized) subset of all 2^N functions on $\{0,1\}^n$. Described by poly(n) long "seed" strings (non-uniform)

Pseudorandom Functions

- Pseudorandom function:
  - A small ($2^{\text{polylog}(N)}$ sized) subset of all 2^N functions on $\{0,1\}^n$. Described by poly(n) long "seed" strings (non-uniform)
  - Each can be evaluated by a poly(n)-size circuit

Pseudorandom Functions

- Pseudorandom function:
  - A small ($2^{\text{polylog}(N)}$ sized) subset of all 2^N functions on $\{0,1\}^n$. Described by poly(n) long "seed" strings (non-uniform)
  - Each can be evaluated by a poly(n)-size circuit
  - A distribution (for each n) defined by uniformly picking a seed

Pseudorandom Functions

- Pseudorandom function:
  - A small ($2^{polylog(N)}$ sized) subset of all 2^N functions on $\{0,1\}^n$. Described by poly(n) long "seed" strings (non-uniform)
  - Each can be evaluated by a poly(n)-size circuit
  - A distribution (for each n) defined by uniformly picking a seed
- Random function: distribution defined by uniformly picking a function (N long string)

Pseudorandom Functions

- Pseudorandom function:
  - A small ($2^{\text{polylog}(N)}$ sized) subset of all 2^N functions on $\{0,1\}^n$. Described by poly(n) long "seed" strings (non-uniform)
  - Each can be evaluated by a poly(n)-size circuit
  - A distribution (for each n) defined by uniformly picking a seed
- Random function: distribution defined by uniformly picking a function (N long string)
- The two are "indistinguishable"

Indistinguishability

Indistinguishability

- Two distributions X and Y are ϵ-indistinguishable to a distinguisher D (which outputs a single bit), if it behaves virtually identically when given samples from either distribution

Indistinguishability

- Two distributions X and Y are ϵ-indistinguishable to a distinguisher D (which outputs a single bit), if it behaves virtually identically when given samples from either distribution
  - $\Pr_{f \leftarrow X}[D(f) \text{ outputs } 1] - \Pr_{f \leftarrow Y}[D(f) \text{ outputs } 1] < \epsilon$

Indistinguishability

- Two distributions X and Y are ϵ-indistinguishable to a distinguisher D (which outputs a single bit), if it behaves virtually identically when given samples from either distribution
  - $\Pr_{f \leftarrow X}[D(f) \text{ outputs } 1] - \Pr_{f \leftarrow Y}[D(f) \text{ outputs } 1] < \epsilon$
- X, Y are ϵ-indistinguishable for size-S distinguishers if this holds for all circuits D of size at most S

(Strong) PRF

(Strong) PRF

- PRF: a distribution over P/poly functions, indistinguishable from random functions

(Strong) PRF

- PRF: a distribution over P/poly functions, indistinguishable from random functions
  - Distinguisher gets the truth-table of the function as input

(Strong) PRF

- PRF: a distribution over P/poly functions, indistinguishable from random functions
  - Distinguisher gets the truth-table of the function as input
- Given any $S(N) = \text{poly}(N)$, can construct a $1/N$-indistinguishable PRF for all size-S distinguishers with seed-length poly(n)

(Strong) PRF

- PRF: a distribution over P/poly functions, indistinguishable from random functions
  - Distinguisher gets the truth-table of the function as input
- Given any $S(N) = poly(N)$, can construct a $1/N$-indistinguishable PRF for all size-S distinguishers with seed-length $poly(n)$
  - If (strong) "one-way functions" exist

(Strong) PRF

- PRF: a distribution over P/poly functions, indistinguishable from random functions
  - Distinguisher gets the truth-table of the function as input
- Given any S(N) = poly(N), can construct a 1/N-indistinguishable PRF for all size-S distinguishers with seed-length poly(n)
  - If (strong) "one-way functions" exist
- (Strong PRF, because "usual" PRF is against poly(**n**)-size distinguishers who can in particular read only poly(n) positions of the truth-table)

Limitations of Natural Proofs

Limitations of Natural Proofs

- PRF: A distribution of P/poly functions, ϵ-indistinguishable from random functions, for $\epsilon < 1/N$, for size-$S(N)$=poly(N) distinguishers D

Limitations of Natural Proofs

- PRF: A distribution of P/poly functions, ϵ-indistinguishable from random functions, for $\epsilon < 1/N$, for size-$S(N)=\text{poly}(N)$ distinguishers D
- A natural property that avoids P/poly can be used to efficiently distinguish a distribution of P/poly functions from random functions

Limitations of Natural Proofs

- PRF: A distribution of P/poly functions, ϵ-indistinguishable from random functions, for $\epsilon < 1/N$, for size-$S(N)$=poly(N) distinguishers D
- A natural property that avoids P/poly can be used to efficiently distinguish a distribution of P/poly functions from random functions
  - Let $D(f)$ be $\Phi(f)$: D of size S = poly (because Φ natural). $Pr_{f \leftarrow all}[\Phi(f)=1] > 1/N$ (because Φ natural), and if Φ avoids P/poly then $Pr_{f \leftarrow PRF}[\Phi(f)=1] = 0$

Limitations of Natural Proofs

- PRF: A distribution of P/poly functions, ϵ-indistinguishable from random functions, for $\epsilon < 1/N$, for size-$S(N)$=poly(N) distinguishers D
- A natural property that avoids P/poly can be used to efficiently distinguish a distribution of P/poly functions from random functions
  - Let $D(f)$ be $\Phi(f)$: D of size S = poly (because Φ natural). $\Pr_{f\leftarrow \text{all}}[\Phi(f)=1] > 1/N$ (because Φ natural), and if Φ avoids P/poly then $\Pr_{f\leftarrow \text{PRF}}[\Phi(f)=1] = 0$
    - Contradiction!

Limitations of Natural Proofs

- PRF: A distribution of P/poly functions, ϵ-indistinguishable from random functions, for $\epsilon < 1/N$, for size-$S(N)=poly(N)$ distinguishers D
- A natural property that avoids P/poly can be used to efficiently distinguish a distribution of P/poly functions from random functions
  - Let $D(f)$ be $\Phi(f)$: D of size $S = poly$ (because Φ natural). $Pr_{f\leftarrow all}[\Phi(f)=1] > 1/N$ (because Φ natural), and if Φ avoids P/poly then $Pr_{f\leftarrow PRF}[\Phi(f)=1] = 0$
    - Contradiction!
- If PRFs exist, then no natural property that avoids P/poly exists

Summary

Summary

- Natural proofs: which use a "natural property" Φ to separate low-complexity functions from high-complexity functions

Summary

- Natural proofs: which use a "natural property" Φ to separate low-complexity functions from high-complexity functions
  - Φ holds for a random function with good probability

Summary

- Natural proofs: which use a "natural property" Φ to separate low-complexity functions from high-complexity functions
  - Φ holds for a random function with good probability
  - Φ is "efficiently" computable

Summary

- Natural proofs: which use a "natural property" Φ to separate low-complexity functions from high-complexity functions
    - Φ holds for a random function with good probability
    - Φ is "efficiently" computable
- Goes against the notion of pseudorandomness

Summary

- Natural proofs: which use a "natural property" Φ to separate low-complexity functions from high-complexity functions
  - Φ holds for a random function with good probability
  - Φ is "efficiently" computable
- Goes against the notion of pseudorandomness
  - Low-complexity functions which look-like a random function, to "efficient" distinguishers

Summary

- Natural proofs: which use a "natural property" Φ to separate low-complexity functions from high-complexity functions
  - Φ holds for a random function with good probability
  - Φ is "efficiently" computable
- Goes against the notion of pseudorandomness
  - Low-complexity functions which look-like a random function, to "efficient" distinguishers
- Natural proofs can't separate out P/poly as low-complexity, if pseudorandom functions exist in P/poly (as we believe)