Lower Bounds in Theory of Computing

Jeff Kinne

Indiana State University, Math and CS Dept.

Math and CS Dept. Seminar, March 21, 2012

Notes

- Pictures on the chalk board (sorry to online viewers...)
- Slides will be online at http://www.kinnejeff.com
- General-purpose links for complexity theory:
  Computational Complexity: A Modern Approach
  lecture notes
  Wikipedia

Goal

What is the smallest running time possible?

Goal

What is the smallest running time possible?

- Requires: **upper bound** and **lower bound**

Goal

What is the smallest running time possible?

- Requires: **upper bound** and **lower bound**

Goal

What is the smallest running time possible?

- Requires: **upper bound** and **lower bound**

Examples

- Addition

Goal

What is the smallest running time possible?

- Requires: **upper bound** and **lower bound**

Examples

- Addition
- Multiplication

Goal

What is the smallest running time possible?

- Requires: **upper bound** and **lower bound**

Examples

- Addition
- Multiplication
- 3-coloring

Goal

What is the smallest running time possible?

- Requires: **upper bound** and **lower bound**

Examples

- Addition
- Multiplication
- 3-coloring
- Factoring

Other Resources/Goals

- Memory space

Other Resources/Goals

- Memory space
- Nondeterminism

Other Resources/Goals

- Memory space
- Nondeterminism
- Communication

Other Resources/Goals

- Memory space
- Nondeterminism
- Communication
- Non-uniformity

Other Resources/Goals

- Memory space
- Nondeterminism
- Communication
- Non-uniformity
- Randomness

Other Resources/Goals

- Memory space
- Nondeterminism
- Communication
- Non-uniformity
- Randomness
- Quantumness

Other Resources/Goals

- Memory space
- Nondeterminism
- Communication
- Non-uniformity
- Randomness
- Quantumness
- ...

Other Resources/Goals

- Memory space
- Nondeterminism
- Communication
- Non-uniformity
- Randomness
- Quantumness
- ...
- Average-case

Other Resources/Goals

- Memory space
- Nondeterminism
- Communication
- Non-uniformity
- Randomness
- Quantumness
- ...
- Average-case , approximation

Other Resources/Goals

- Memory space
- Nondeterminism
- Communication
- Non-uniformity
- Randomness
- Quantumness
- ...
- Average-case , approximation

- See, e.g., the the "Complexity Zoo"

Why the Zoo of Complexity Classes?

- Diverse goals in the world

Why the Zoo of Complexity Classes?

- Diverse goals in the world
- Class captures important/interesting problems – e.g. NP

NP

P versus NP problem

P versus NP problem

If P = NP...

P versus NP problem

If P = NP...

- Perfect optimization

P versus NP problem

If P = NP...

- Perfect optimization
- Computer search to prove unknown conjectures

P versus NP problem

If P = NP...

- Perfect optimization
- Computer search to prove unknown conjectures
- No cryptography/encryption

P versus NP problem

If P = NP...

- Perfect optimization
- Computer search to prove unknown conjectures
- No cryptography/encryption (see one-way functions, RSA)

P versus NP problem

If P = NP...

- Perfect optimization
- Computer search to prove unknown conjectures
- No cryptography/encryption (see one-way functions, RSA)

If P $\neq$ NP...

P versus NP problem

If P = NP...

- Perfect optimization
- Computer search to prove unknown conjectures
- No cryptography/encryption (see one-way functions, RSA)

If P $\neq$ NP...

- Cannot approximate some optimization problems

P versus NP problem

If P = NP...

- Perfect optimization
- Computer search to prove unknown conjectures
- No cryptography/encryption (see one-way functions, RSA)

If P $\neq$ NP...

- Cannot approximate some optimization problems (PCP Theorem – "randomized" proofs)

P versus NP problem

If P = NP...

- Perfect optimization
- Computer search to prove unknown conjectures
- No cryptography/encryption (see one-way functions, RSA)

If P $\neq$ NP...

- Cannot approximate some optimization problems (PCP Theorem – “randomized” proofs)
- Need more to get cryptography

P versus NP problem

If P = NP...

- Perfect optimization
- Computer search to prove unknown conjectures
- No cryptography/encryption (see one-way functions, RSA)

If P $\neq$ NP...

- Cannot approximate some optimization problems (PCP Theorem – "randomized" proofs)
- Need more to get cryptography
- NP still could be "normally" easy

Definition

NTIME(t) – guess t size certificate

Definition

NTIME(t) – guess t size certificate

Trivial Upper Bound

NTIME(t) can be solved in $2^{O(t)}$ time.

Definition

NTIME(t) – guess t size certificate

Trivial Upper Bound

NTIME(t) can be solved in $2^{O(t)}$ time.

Slightly better, e.g., 3-coloring

- $\sum_{k=0}^{n} \binom{n}{k} k^2 3^{k/3} \leq n^2 \sum_{k=0}^{n} \binom{n}{k} 3^{k/3} = n^2(1+3^{1/3})^n$

Definition

NTIME(t) – guess t size certificate

Trivial Upper Bound

NTIME(t) can be solved in $2^{O(t)}$ time.

Slightly better, e.g., 3-coloring

- $\sum_{k=0}^{n} \binom{n}{k} k^2 3^{k/3} \leq n^2 \sum_{k=0}^{n} \binom{n}{k} 3^{k/3} = n^2(1+3^{1/3})^n$
- Number of **maximal independent sets** is at most $3^{n/3}$.

Definition

NTIME(t) – guess t size certificate

Trivial Upper Bound

NTIME(t) can be solved in $2^{O(t)}$ time.

Slightly better, e.g., 3-coloring

- $\sum_{k=0}^{n} \binom{n}{k} k^2 3^{k/3} \leq n^2 \sum_{k=0}^{n} \binom{n}{k} 3^{k/3} = n^2(1+3^{1/3})^n$
- Number of **maximal independent sets** is at most $3^{n/3}$.
- Look at all subgraphs G_S from smallest to largest

Definition

NTIME(t) – guess t size certificate

Trivial Upper Bound

NTIME(t) can be solved in $2^{O(t)}$ time.

Slightly better, e.g., 3-coloring

- $\sum_{k=0}^{n} \binom{n}{k} k^2 3^{k/3} \leq n^2 \sum_{k=0}^{n} \binom{n}{k} 3^{k/3} = n^2(1+3^{1/3})^n$
- Number of **maximal independent sets** is at most $3^{n/3}$.
- Look at all subgraphs G_S from smallest to largest
- $\mathrm{OPT}(G_S) = 1 + \min(\mathrm{OPT}(G_{S-T})$ — T a max ind set in G_S).

Definition

NTIME(t) – guess t size certificate

Trivial Upper Bound

NTIME(t) can be solved in $2^{O(t)}$ time.

Slightly better, e.g., 3-coloring

- $\sum_{k=0}^{n} \binom{n}{k} k^2 3^{k/3} \leq n^2 \sum_{k=0}^{n} \binom{n}{k} 3^{k/3} = n^2(1+3^{1/3})^n$
- Number of **maximal independent sets** is at most $3^{n/3}$.
- Look at all subgraphs G_S from smallest to largest
- $\mathrm{OPT}(G_S) = 1 + \min(\mathrm{OPT}(G_{S-T})$ — T a max ind set in G_S).

survey on exact NP-complete algorithms

Exponential Time Hypothesis

3SAT (and some other NP-complete problems)
cannot be decided in time $2^{\epsilon n}$ time for some $\epsilon > 0$.

Exponential Time Hypothesis

3SAT (and some other NP-complete problems) cannot be decided in time $2^{\epsilon n}$ time for some $\epsilon > 0$.

- Not true for 3-coloring.

Exponential Time Hypothesis

3SAT (and some other NP-complete problems) cannot be decided in time $2^{\epsilon n}$ time for some $\epsilon > 0$.

- Not true for 3-coloring.
- How close are we to proving this?

Exponential Time Hypothesis

3SAT (and some other NP-complete problems) cannot be decided in time $2^{\epsilon n}$ time for some $\epsilon > 0$.

- Not true for 3-coloring.
- How close are we to proving this?
- Undecidable problems – e.g. Halting Problem

Exponential Time Hypothesis

3SAT (and some other NP-complete problems) cannot be decided in time $2^{\epsilon n}$ time for some $\epsilon > 0$.

- Not true for 3-coloring.
- How close are we to proving this?
- Undecidable problems – e.g. Halting Problem
- Almost all decision problems are undecidable.

Exponential Time Hypothesis

3SAT (and some other NP-complete problems) cannot be decided in time $2^{\epsilon n}$ time for some $\epsilon > 0$.

- Not true for 3-coloring.
- How close are we to proving this?
- Undecidable problems – e.g. Halting Problem
- Almost all decision problems are undecidable.
- Smallest class known to require 2^n time?

Exponential Time Hypothesis

3SAT (and some other NP-complete problems) cannot be decided in time $2^{\epsilon n}$ time for some $\epsilon > 0$.

- Not true for 3-coloring.
- How close are we to proving this?
- Undecidable problems – e.g. Halting Problem
- Almost all decision problems are undecidable.
- Smallest class known to require 2^n time? ...

Exponential Time Hypothesis

3SAT (and some other NP-complete problems)
cannot be decided in time $2^{\epsilon n}$ time for some $\epsilon > 0$.

- Not true for 3-coloring.
- How close are we to proving this?
- Undecidable problems – e.g. Halting Problem
- Almost all decision problems are undecidable.
- Smallest class known to require 2^n time? ... Exponential Time

Exponential Time Hypothesis

3SAT (and some other NP-complete problems) cannot be decided in time $2^{\epsilon n}$ time for some $\epsilon > 0$.

- Not true for 3-coloring.
- How close are we to proving this?
- Undecidable problems – e.g. Halting Problem
- Almost all decision problems are undecidable.
- Smallest class known to require 2^n time? ... Exponential Time (diagonalization...)

Exponential Time Hypothesis

3SAT (and some other NP-complete problems) cannot be decided in time $2^{\epsilon n}$ time for some $\epsilon > 0$.

- Not true for 3-coloring.
- How close are we to proving this?
- Undecidable problems – e.g. Halting Problem
- Almost all decision problems are undecidable.
- Smallest class known to require 2^n time? ... Exponential Time (diagonalization...)
- It could be that 3SAT is in $O(n)$ time.

Theorem

SAT cannot be solved in simultaneous time n^c *and space* n^d *when* $c \cdot (c + d) < 2$.

Theorem

SAT cannot be solved in simultaneous time n^c and space n^d when $c \cdot (c + d) < 2$.

survey on similar results

Theorem

SAT cannot be solved in simultaneous time n^c and space n^d when $c \cdot (c + d) < 2$.

survey on similar results

- Definition: $\mathrm{NTIME}(n^2)$ – guess $O(n^2)$ size certificate

Theorem

SAT cannot be solved in simultaneous time n^c and space n^d when $c \cdot (c + d) < 2$.

survey on similar results

- Definition: $\mathrm{NTIME}(n^2)$ – guess $O(n^2)$ size certificate
- If theorem false...

Theorem

SAT cannot be solved in simultaneous time n^c and space n^d when $c \cdot (c + d) < 2$.

survey on similar results

- Definition: NTIME(n^2) – guess $O(n^2)$ size certificate
- If theorem false...
- NTIME(n^2) $\subseteq$ time n^{2c}, space n^{2d}

Theorem

SAT cannot be solved in simultaneous time n^c and space n^d when $c \cdot (c + d) < 2$.

survey on similar results

- Definition: $\mathrm{NTIME}(n^2)$ – guess $O(n^2)$ size certificate
- If theorem false...
- $\mathrm{NTIME}(n^2) \subseteq$ time n^{2c}, space n^{2d}
- $\subseteq \exists \, \forall \, \mathrm{TIME}(n^{c+d})$

Theorem

SAT cannot be solved in simultaneous time n^c and space n^d when $c \cdot (c + d) < 2$.

survey on similar results

- Definition: $\mathrm{NTIME}(n^2)$ – guess $O(n^2)$ size certificate
- If theorem false...
- $\mathrm{NTIME}(n^2) \subseteq$ time n^{2c}, space n^{2d}
- $\subseteq \exists\, \forall\, \mathrm{TIME}(n^{c+d})$
- $\subseteq \mathrm{NTIME}(n^{c \cdot (c+d)})$

Theorem

SAT cannot be solved in simultaneous time n^c and space n^d when $c \cdot (c + d) < 2$.

survey on similar results

- Definition: $\mathrm{NTIME}(n^2)$ – guess $O(n^2)$ size certificate
- If theorem false...
- $\mathrm{NTIME}(n^2) \subseteq$ time n^{2c}, space n^{2d}
- $\subseteq \exists \, \forall \, \mathrm{TIME}(n^{c+d})$
- $\subseteq \mathrm{NTIME}(n^{c \cdot (c+d)})$
- Contradiction if $2 > c \cdot (c + d)$

Exponential Lower Bounds

Parity

Is number of 1's in binary string even or odd?

Parity

Is number of 1's in binary string even or odd?

Theorem (Hastad)

A depth d circuit for parity has size at least $2^{\epsilon \cdot n^{1/(d-1)}}$ for some constant $\epsilon > 0$.

Parity

Is number of 1's in binary string even or odd?

Theorem (Hastad)

A depth d circuit for parity has size at least $2^{\epsilon \cdot n^{1/(d-1)}}$ for some constant $\epsilon > 0$.

Theorem (Razborov-Smolensky)

Same as above, but size is $2^{\epsilon \cdot n^{1/(2d)}}$.

Parity

Is number of 1's in binary string even or odd?

Theorem (Hastad)

A depth d circuit for parity has size at least $2^{\epsilon \cdot n^{1/(d-1)}}$ for some constant $\epsilon > 0$.

Theorem (Razborov-Smolensky)

Same as above, but size is $2^{\epsilon \cdot n^{1/(2d)}}$.

The Complexity of Finite Functions, Boppana and Sipser

Theorem (Razborov-Smolensky)

A depth d circuit for parity has size at least $2^{\epsilon \cdot n^{1/(2d)}}$ for some constant $\epsilon > 0$.

Theorem (Razborov-Smolensky)

A depth d circuit for parity has size at least $2^{\epsilon \cdot n^{1/(2d)}}$ for some constant $\epsilon > 0$.

- Depth d, size S circuit

Theorem (Razborov-Smolensky)

A depth d circuit for parity has size at least $2^{\epsilon \cdot n^{1/(2d)}}$ for some constant $\epsilon > 0$.

- Depth d, size S circuit
- $\Rightarrow$ degree $\sqrt{n}$ poly, makes at most $2^n \cdot \frac{S}{2^{n^{1/(2d)}/2}}$ mistakes

Theorem (Razborov-Smolensky)

A depth d circuit for parity has size at least $2^{\epsilon \cdot n^{1/(2d)}}$ for some constant $\epsilon > 0$.

- Depth d, size S circuit
- $\Rightarrow$ degree $\sqrt{n}$ poly, makes at most $2^n \cdot \frac{S}{2^{n^{1/(2d)}/2}}$ mistakes
- Any $\sqrt{n}$-degree poly makes at least $2^n \cdot \frac{1}{50}$ mistakes

"Enhanced" constant-depth circuits

"Enhanced" constant-depth circuits

- Allow more gates than just AND, OR, NOT

"Enhanced" constant-depth circuits

- Allow more gates than just AND, OR, NOT
- mod p, parity, majority

"Enhanced" constant-depth circuits

- Allow more gates than just AND, OR, NOT
- mod p, parity, majority
- Intermediate between constant-depth and not

"Enhanced" constant-depth circuits

- Allow more gates than just AND, OR, NOT
- mod p, parity, majority
- Intermediate between constant-depth and not

Theorem (Allender, ..., Kinne)

Uniform *depth d circuits* **with majority gates** *for* **matrix permanent** *have size at least* $S(n)$,

"Enhanced" constant-depth circuits

- Allow more gates than just AND, OR, NOT
- mod p, parity, majority
- Intermediate between constant-depth and not

Theorem (Allender, ..., Kinne)

Uniform *depth d circuits* **with majority gates** *for* **matrix permanent** *have size at least* $S(n)$,
for $S(n)$ *that satisfy* $S^{(O(d))}(n) < 2^n$.

Theorem (Allender, ..., Kinne)

Uniform *depth d circuits* **with majority gates** *for* **matrix permanent** *have size at least $S(n)$,*
for $S(n)$ that satisfy $S^{(O(d))}(n) < 2^n$.

Theorem (Allender, ..., Kinne)

Uniform *depth d circuits* **with majority gates** *for* **matrix permanent** *have size at least $S(n)$,*
for $S(n)$ that satisfy $S^{(O(d))}(n) < 2^n$.

- "Hard" problem H in EXP requires size 2^n (*uniform*) circuits

Theorem (Allender, ..., Kinne)

Uniform *depth d circuits* **with majority gates** *for* **matrix permanent** *have size at least $S(n)$, for $S(n)$ that satisfy $S^{(O(d))}(n) < 2^n$.*

- "Hard" problem H in EXP requires size 2^n (*uniform*) circuits
- Assume depth d, size $S(n)$ circuits for permanent

Theorem (Allender, ..., Kinne)

Uniform *depth d circuits* **with majority gates** *for* **matrix permanent** *have size at least $S(n)$, for $S(n)$ that satisfy $S^{(O(d))}(n) < 2^n$.*

- "Hard" problem H in EXP requires size 2^n (*uniform*) circuits
- Assume depth d, size $S(n)$ circuits for permanent
- $\Rightarrow$ size $\approx S(2^n)$, depth d circuit C for H

Theorem (Allender, ..., Kinne)

Uniform *depth d circuits* **with majority gates** *for* **matrix permanent** *have size at least $S(n)$,*
for $S(n)$ that satisfy $S^{(O(d))}(n) < 2^n$.

- "Hard" problem H in EXP requires size 2^n (*uniform*) circuits
- Assume depth d, size $S(n)$ circuits for permanent
- $\Rightarrow$ size $\approx S(2^n)$, depth d circuit C for H
- Bottom majority gates in C $\Rightarrow$

Theorem (Allender, ..., Kinne)

Uniform *depth d circuits* **with majority gates** *for* **matrix permanent** *have size at least* $S(n)$,
for $S(n)$ *that satisfy* $S^{(O(d))}(n) < 2^n$.

- "Hard" problem H in EXP requires size 2^n (*uniform*) circuits
- Assume depth d, size $S(n)$ circuits for permanent
- $\Rightarrow$ size $\approx S(2^n)$, depth d circuit C for H
- Bottom majority gates in $C \Rightarrow$
  permanent question of size $\approx \log(S(2^n)) + n$

Theorem (Allender, ..., Kinne)

Uniform *depth d circuits* **with majority gates** *for* **matrix permanent** *have size at least* $S(n)$,
for $S(n)$ *that satisfy* $S^{(O(d))}(n) < 2^n$.

- "Hard" problem H in EXP requires size 2^n (*uniform*) circuits
- Assume depth d, size $S(n)$ circuits for permanent
- $\Rightarrow$ size $\approx S(2^n)$, depth d circuit C for H
- Bottom majority gates in $C \Rightarrow$
  permanent question of size $\approx \log(S(2^n)) + n$
  size $S_1 = S(\log(S(2^n)) + n)$ circuit

Theorem (Allender, ..., Kinne)

Uniform *depth d circuits* **with majority gates** *for* **matrix permanent** *have size at least* $S(n)$,
for $S(n)$ *that satisfy* $S^{(O(d))}(n) < 2^n$.

- "Hard" problem H in EXP requires size 2^n (*uniform*) circuits
- Assume depth d, size $S(n)$ circuits for permanent
- $\Rightarrow$ size $\approx S(2^n)$, depth d circuit C for H
- Bottom majority gates in $C \Rightarrow$
  permanent question of size $\approx \log(S(2^n)) + n$
  size $S_1 = S(\log(S(2^n)) + n)$ circuit
- Next level of majority gates $\Rightarrow$

Theorem (Allender, ..., Kinne)

Uniform *depth d circuits* **with majority gates** *for* **matrix permanent** *have size at least $S(n)$,*
for $S(n)$ that satisfy $S^{(O(d))}(n) < 2^n$.

- "Hard" problem H in EXP requires size 2^n (*uniform*) circuits
- Assume depth d, size $S(n)$ circuits for permanent
- $\Rightarrow$ size $\approx S(2^n)$, depth d circuit C for H
- Bottom majority gates in C $\Rightarrow$
  permanent question of size $\approx \log(S(2^n)) + n$
  size $S_1 = S(\log(S(2^n)) + n)$ circuit
- Next level of majority gates $\Rightarrow$
  permanent question of size $\approx \log(S(2^n)) + n + S_1$

Theorem (Allender, ..., Kinne)

Uniform *depth d circuits* **with majority gates** *for* **matrix permanent** *have size at least* $S(n)$,
for $S(n)$ *that satisfy* $S^{(O(d))}(n) < 2^n$.

- "Hard" problem H in EXP requires size 2^n (*uniform*) circuits
- Assume depth d, size $S(n)$ circuits for permanent
- $\Rightarrow$ size $\approx S(2^n)$, depth d circuit C for H
- Bottom majority gates in $C \Rightarrow$
  permanent question of size $\approx \log(S(2^n)) + n$
  size $S_1 = S(\log(S(2^n)) + n)$ circuit
- Next level of majority gates $\Rightarrow$
  permanent question of size $\approx \log(S(2^n)) + n + S_1$

  ...

To Conclude...