# BILL AND NATHAN RECORD LECTURE!!!!

BILL AND NATHAN RECORD LECTURE!!!

# Other Topics I Could Have Covered And Might Next Spring

May 3, 2024

# Other Topics I Could Have Covered And Might Next Spring

Exposition by William Gasarch—U of MD

# Steps Forward and Backwards

Complexity theory has its roots in recursion theory.

# Steps Forward and Backwards

Complexity theory has its roots in recursion theory.

However, over the last 40 years research in complexity theory has drawn less and less on logic and more and more on combinatorics.

# Steps Forward and Backwards

Complexity theory has its roots in recursion theory.

However, over the last 40 years research in complexity theory has drawn less and less on logic and more and more on combinatorics.

**A Step Forward** means a topic that will help modernize the course. Perhaps any result after 1990.

# Steps Forward and Backwards

Complexity theory has its roots in recursion theory.

However, over the last 40 years research in complexity theory has drawn less and less on logic and more and more on combinatorics.

**A Step Forward** means a topic that will help modernize the course. Perhaps any result after 1990.

**A Step Backwards** means an old topic, we'll say pre-1980. Logic or more tied to the actual machine model. This is not necc bad.

# Topics on Reg Langs

Exposition by William Gasarch—U of MD

# How Reg Langs are Really Used

# How Reg Langs are Really Used

1. Pattern Matching

# How Reg Langs are Really Used

1. Pattern Matching
2. Perl-Regular, Ruby-Regular, etc.

# How Reg Langs are Really Used

1. Pattern Matching
2. Perl-Regular, Ruby-Regular, etc.
3. Using DFA's to model systems

# How Reg Langs are Really Used

1. Pattern Matching
2. Perl-Regular, Ruby-Regular, etc.
3. Using DFA's to model systems
4. Alg to minimize DFAs

# How Reg Langs are Really Used

1. Pattern Matching
2. Perl-Regular, Ruby-Regular, etc.
3. Using DFA's to model systems
4. Alg to minimize DFAs

**Verdict** Have not done. Perl-Regular might drive me nuts since it does not have a clean mathematical semantics.

# Desc of Reg Expressions

Theorems about lower bounds on lengths of Regular Expressions.

# Desc of Reg Expressions

Theorems about lower bounds on lengths of Regular Expressions.
**Verdict** Would have to learn those theorems, which I want to.
`https://www.cs.umd.edu/users/gasarch/TOPICS/desc/desc.html`

# Desc of Reg Expressions

Theorems about lower bounds on lengths of Regular Expressions.
**Verdict** Would have to learn those theorems, which I want to.
`https://www.cs.umd.edu/users/gasarch/TOPICS/desc/desc.html`
Goes with the **Length of Description** theme I've had this year.

# Topics on CFL's

Exposition by William Gasarch—U of MD

# Applications

# Applications

1. PDA's are DFA's with a stack and are use to model compilers.

# Applications

1. PDA's are DFA's with a stack and are use to model compilers.
2. Applications of CFG's and PDA's to Compiler design

# Applications

1. PDA's are DFA's with a stack and are use to model compilers.
2. Applications of CFG's and PDA's to Compiler design
3. C++ syntax is undecidable

# Applications

1. PDA's are DFA's with a stack and are use to model compilers.
2. Applications of CFG's and PDA's to Compiler design
3. C++ syntax is undecidable

**Verdict** Won't be covering. Too messy.

# Desc of Langs

All papers on this are here:

https://www.cs.umd.edu/users/gasarch/TOPICS/desc/desc.html

# Desc of Langs

All papers on this are here:
`https://www.cs.umd.edu/users/gasarch/TOPICS/desc/desc.html`

1. Deterministic PDA's which play into length of descriptions.

# Desc of Langs

All papers on this are here:
`https://www.cs.umd.edu/users/gasarch/TOPICS/desc/desc.html`

1. Deterministic PDA's which play into length of descriptions.
2. $\{ww : |w| = n\}$ requires exp sized Chomsky Norm Form CFG.

# Desc of Langs

All papers on this are here:
https://www.cs.umd.edu/users/gasarch/TOPICS/desc/desc.html

1. Deterministic PDA's which play into length of descriptions.
2. $\{ww : |w| = n\}$ requires exp sized Chomsky Norm Form CFG.
3. Show that $\mathrm{CFG}\Sigma^* \equiv_T \mathrm{INF}$.

# Desc of Langs

All papers on this are here:
`https://www.cs.umd.edu/users/gasarch/TOPICS/desc/desc.html`

1. Deterministic PDA's which play into length of descriptions.
2. $\{ww : |w| = n\}$ requires exp sized Chomsky Norm Form CFG.
3. Show that $\mathrm{CFG}\Sigma^* \equiv_T \mathrm{INF}$.
4. Show that the bding funct for CFG-REG is $\equiv_T \mathrm{INF}$.

# Desc of Langs

All papers on this are here:
`https://www.cs.umd.edu/users/gasarch/TOPICS/desc/desc.html`

1. Deterministic PDA's which play into length of descriptions.
2. $\{ww : |w| = n\}$ requires exp sized Chomsky Norm Form CFG.
3. Show that $\mathrm{CFG}\Sigma^* \equiv_T \mathrm{INF}$.
4. Show that the bding funct for CFG-REG is $\equiv_T \mathrm{INF}$.

**Verdict** A Bill-topic. Seems backward looking.

# Desc of Langs

All papers on this are here:
`https://www.cs.umd.edu/users/gasarch/TOPICS/desc/desc.html`

1. Deterministic PDA's which play into length of descriptions.
2. $\{ww : |w| = n\}$ requires exp sized Chomsky Norm Form CFG.
3. Show that $\mathrm{CFG}\Sigma^* \equiv_T \mathrm{INF}$.
4. Show that the bding funct for CFG-REG is $\equiv_T \mathrm{INF}$.

**Verdict** A Bill-topic. Seems backward looking.
But goes with the **Size of Device** theme.

# Topics on Complexity Theory

Exposition by William Gasarch—U of MD

# More RESPECT

**Recall** **RESPECT** is shorthand for **Lower Bounds are Hard because you never know when someone will come along with clever math or deep math or SOMETHING that your so-called lower bound did not take into account**.

# More RESPECT

**Recall** **RESPECT** is shorthand for **Lower Bounds are Hard because you never know when someone will come along with clever math or deep math or SOMETHING that your so-called lower bound did not take into account**.

1. From the Graph Minor Theorem one obtains MANY problems in P. RESPECT

# More RESPECT

**Recall RESPECT** is shorthand for **Lower Bounds are Hard because you never know when someone will come along with clever math or deep math or SOMETHING that your so-called lower bound did not take into account**.

1. From the Graph Minor Theorem one obtains MANY problems in P. RESPECT
2. Hall's Matching Theorem leads to a particular SAT-type problem being in P. RESPECT

# More RESPECT

**Recall RESPECT** is shorthand for **Lower Bounds are Hard because you never know when someone will come along with clever math or deep math or SOMETHING that your so-called lower bound did not take into account**.

1. From the Graph Minor Theorem one obtains MANY problems in P. RESPECT
2. Hall's Matching Theorem leads to a particular SAT-type problem being in P. RESPECT
3. SAT Solvers- while not in P, do surprisingly well. RESPECT

# More RESPECT

**Recall RESPECT** is shorthand for **Lower Bounds are Hard because you never know when someone will come along with clever math or deep math or SOMETHING that your so-called lower bound did not take into account**.

1. From the Graph Minor Theorem one obtains MANY problems in P. RESPECT
2. Hall's Matching Theorem leads to a particular SAT-type problem being in P. RESPECT
3. SAT Solvers- while not in P, do surprisingly well. RESPECT

**Verdict** I should write a parody of Aretha Franklin's song RESPECT with this theme.

# More RESPECT

**Recall RESPECT** is shorthand for **Lower Bounds are Hard because you never know when someone will come along with clever math or deep math or SOMETHING that your so-called lower bound did not take into account**.

1. From the Graph Minor Theorem one obtains MANY problems in P. RESPECT
2. Hall's Matching Theorem leads to a particular SAT-type problem being in P. RESPECT
3. SAT Solvers- while not in P, do surprisingly well. RESPECT

**Verdict** I should write a parody of Aretha Franklin's song RESPECT with this theme.
Also, would be happy to do any of these topics.

# SEND+MORE=MONEY

$$\begin{array}{ccccc} & S & E & N & D \\ + & M & O & R & E \\ \hline M & O & N & E & Y \end{array}$$

# SEND+MORE=MONEY

$$
\begin{array}{ccccc}
 & S & E & N & D \\
+ & M & O & R & E \\
\hline
M & O & N & E & Y
\end{array}
$$

Has Solution

$$
\begin{array}{ccccc}
 & 9 & 5 & 6 & 7 \\
+ & 1 & 0 & 8 & 5 \\
\hline
1 & 0 & 6 & 5 & 2
\end{array}
$$

# SEND+MORE=MONEY

| | | | | |
|---|---|---|---|---|
| | S | E | N | D |
| + | M | O | R | E |
| M | O | N | E | Y |

Has Solution

| | | | | |
|---|---|---|---|---|
| | 9 | 5 | 6 | 7 |
| + | 1 | 0 | 8 | 5 |
| 1 | 0 | 6 | 5 | 2 |

Given a puzzle, does it have a solution, is NP-complete

# SEND+MORE=MONEY

| | | | | |
|---|---|---|---|---|
| | S | E | N | D |
| + | M | O | R | E |
| M | O | N | E | Y |

Has Solution

| | | | | |
|---|---|---|---|---|
| | 9 | 5 | 6 | 7 |
| + | 1 | 0 | 8 | 5 |
| 1 | 0 | 6 | 5 | 2 |

Given a puzzle, does it have a solution, is NP-complete

**Verdict** Not sure. Good to see one hard reduction. Too hard?

# Other Problems that are Hard

# Other Problems that are Hard

1. CHESS is EXPTIME-complete

# Other Problems that are Hard

1. CHESS is EXPTIME-complete
2. GO is EXPTIME-complete

# Other Problems that are Hard

1. CHESS is EXPTIME-complete
2. GO is EXPTIME-complete
3. Equiv of trex is EXPSPACE-complete

# Other Problems that are Hard

1. CHESS is EXPTIME-complete
2. GO is EXPTIME-complete
3. Equiv of trex is EXPSPACE-complete
4. Powerpoint and LaTeX are undecidable.

# Other Problems that are Hard

1. CHESS is EXPTIME-complete
2. GO is EXPTIME-complete
3. Equiv of trex is EXPSPACE-complete
4. Powerpoint and LaTeX are undecidable.

**Verdict** Interesting Results but Messy Proofs.

# Other Problems that are Hard

1. CHESS is EXPTIME-complete
2. GO is EXPTIME-complete
3. Equiv of trex is EXPSPACE-complete
4. Powerpoint and LaTeX are undecidable.

**Verdict** Interesting Results but Messy Proofs.
Perhaps should define EXPTIME-complete so can STATE these results.

# Lower Bounds on Approx

**Thm** If there is an algorithm that, given an instance of TSP, outputs a cycle that is $\leq 10\mathrm{OPT}$, then $\mathrm{P} = \mathrm{NP}$.

# Lower Bounds on Approx

**Thm** If there is an algorithm that, given an instance of TSP, outputs a cycle that is $\leq 10\mathrm{OPT}$, then $\mathrm{P} = \mathrm{NP}$.
(can replace 10 with any constant).

# Lower Bounds on Approx

**Thm** If there is an algorithm that, given an instance of TSP, outputs a cycle that is $\le 10\mathrm{OPT}$, then $\mathrm{P} = \mathrm{NP}$.
(can replace 10 with any constant).
**Verdict** Meant to do that one this year but forgot. Oh well. Will do it next year.
**Caveat** There are other similar results I could look into.

# Why we think GI is Not NPC

**Thm** If GI is NPC then $\Sigma_2^p = \Pi_2^p$.

# Why we think GI is Not NPC

**Thm** If GI is NPC then $\Sigma_2^p = \Pi_2^p$.

1. Involves some probability

# Why we think GI is Not NPC

**Thm** If GI is NPC then $\Sigma_2^p = \Pi_2^p$.

1. Involves some probability
2. Would take 2 or 3 lectures.

# Why we think GI is Not NPC

**Thm** If GI is NPC then $\Sigma_2^p = \Pi_2^p$.

1. Involves some probability
2. Would take 2 or 3 lectures.
3. Is very interesting.

# Why we think GI is Not NPC

**Thm** If GI is NPC then $\Sigma_2^p = \Pi_2^p$.

1. Involves some probability
2. Would take 2 or 3 lectures.
3. Is very interesting.

**Verdict** A Step Forward! Might be to hard.

# Sparse Sets

**Def** A set $A$ is **sparse** if $\exists$ poly $p$, $|A \cap \Sigma^n| \leq p(n)$.

# Sparse Sets

**Def** A set $A$ is **sparse** if $\exists$ poly $p$, $|A \cap \Sigma^n| \le p(n)$.

1. **Thm** If a sparse set is NP-complete then P=NP.

# Sparse Sets

**Def** A set $A$ is **sparse** if $\exists$ poly $p$, $|A \cap \Sigma^n| \leq p(n)$.

1. **Thm** If a sparse set is NP-complete then P=NP.
2. **Thm** If a sparse set is NP-hard under poly-Turing reductions then $\Sigma_2^p = \Pi_2^p$.

**Verdict** I have done the first one before. Could do the second. A tiny step backwards.

# There Exists Decidable Sets that are Hard

**Thm** There exists a decidable set $A$ such that
$A \notin \mathrm{DTIME}(ACK(n))$.

# There Exists Decidable Sets that are Hard

**Thm** There exists a decidable set $A$ such that
$A \notin \mathrm{DTIME}(ACK(n))$.
(Can replace $\mathrm{ACK}(n)$ with any computable function.)

# There Exists Decidable Sets that are Hard

**Thm** There exists a decidable set $A$ such that
$A \notin \mathrm{DTIME}(ACK(n))$.
(Can replace $\mathrm{ACK}(n)$ with any computable function.)

Does my Darling care?

# There Exists Decidable Sets that are Hard

**Thm** There exists a decidable set $A$ such that
$A \notin \mathrm{DTIME}(ACK(n))$.
(Can replace $\mathrm{ACK}(n)$ with any computable function.)

Does my Darling care?
**BILL:** Darling, do you find the following result interesting: there is a set $A$ such that $A \notin \mathrm{DTIME}(\mathrm{ACK}(n))$.

# There Exists Decidable Sets that are Hard

**Thm** There exists a decidable set $A$ such that
$A \notin \mathrm{DTIME}(ACK(n))$.
(Can replace $\mathrm{ACK}(n)$ with any computable function.)

Does my Darling care?
**BILL:** Darling, do you find the following result interesting: there is a set $A$ such that $A \notin \mathrm{DTIME}(\mathrm{ACK}(n))$.
**DARLING:** Yes I would care unless—

# There Exists Decidable Sets that are Hard

**Thm** There exists a decidable set $A$ such that
$A \notin \mathrm{DTIME}(ACK(n))$.
(Can replace $\mathrm{ACK}(n)$ with any computable function.)

Does my Darling care?
**BILL:** Darling, do you find the following result interesting: there is a set $A$ such that $A \notin \mathrm{DTIME}(\mathrm{ACK}(n))$.
**DARLING:** Yes I would care unless—
**BILL:** (cuts her off) Great!. Oh, unless what?

# There Exists Decidable Sets that are Hard

**Thm** There exists a decidable set $A$ such that
$A \notin \mathrm{DTIME}(ACK(n))$.
(Can replace $\mathrm{ACK}(n)$ with any computable function.)

Does my Darling care?
**BILL:** Darling, do you find the following result interesting: there is a set $A$ such that $A \notin \mathrm{DTIME}(\mathrm{ACK}(n))$.
**DARLING:** Yes I would care unless—
**BILL:** (cuts her off) Great!. Oh, unless what?
**DARLING:** Unless its one of those dumb-ass set that people like you construct for the **sole purpose** of having that property.

# There Exists Decidable Sets that are Hard

**Thm** There exists a decidable set $A$ such that
$A \notin \mathrm{DTIME}(ACK(n))$.
(Can replace $\mathrm{ACK}(n)$ with any computable function.)

Does my Darling care?
**BILL:** Darling, do you find the following result interesting: there is a set $A$ such that $A \notin \mathrm{DTIME}(\mathrm{ACK}(n))$.
**DARLING:** Yes I would care unless—
**BILL:** (cuts her off) Great!. Oh, unless what?
**DARLING:** Unless its one of those dumb-ass set that people like you construct for the **sole purpose** of having that property.
**BILL:** You nailed it!

# Theorems from Space Complexity

1. Nondet-Log-Space is closed under complement. Good for **Respect** theme.

# Theorems from Space Complexity

1. Nondet-Log-Space is closed under complement. Good for **Respect** theme.
2. Nondet-Log-Space is contained in P. Good for **Respect** theme.

# Theorems from Space Complexity

1. Nondet-Log-Space is closed under complement. Good for **Respect** theme.
2. Nondet-Log-Space is contained in P. Good for **Respect** theme.
3. $\mathrm{NSPACE}(S(n)) \subseteq \mathrm{DSPACE}(S(n)^2)$.

# Theorems from Space Complexity

1. Nondet-Log-Space is closed under complement. Good for **Respect** theme.
2. Nondet-Log-Space is contained in P. Good for **Respect** theme.
3. $\mathrm{NSPACE}(S(n)) \subseteq \mathrm{DSPACE}(S(n)^2)$.

**Verdict** All nice theorems that I could do. Would need to introduce and talk about space complexity so this would take time. Not that hard, so thats good.

# Decidable and Undecidable

Exposition by William Gasarch—U of MD

# Decidable Theories

# Decidable Theories

1. Presburger Arithmetic is decidable: just $<$ and $+$ over $\mathbb{N}$.

# Decidable Theories

1. Presburger Arithmetic is decidable: just $<$ and $+$ over $\mathbb{N}$.
2. Theory of the reals is decidable!

# Decidable Theories

1. Presburger Arithmetic is decidable: just $<$ and $+$ over $\mathbb{N}$.
2. Theory of the reals is decidable!

**Verdict** A step Backwards.

# Arithmetic Hierarchy

Actually **prove** that (say)

$$\mathrm{INF} = \{e : M_e \text{ halts on an infinite number of numbers}\}$$

is NOT in $\Sigma_2$.

# Arithmetic Hierarchy

Actually **prove** that (say)

$$\mathrm{INF} = \{e : M_e \text{ halts on an infinite number of numbers}\}$$

is NOT in $\Sigma_2$.
**Verdict** Too much background and a step backwards.

# Intermediary Sets

Are there sets that are both

# Intermediary Sets

Are there sets that are both

1. Not decidable

# Intermediary Sets

Are there sets that are both

1. Not decidable
2. Weaker than HALT.

# Intermediary Sets

Are there sets that are both

1. Not decidable
2. Weaker than HALT.

Answer: YES and the proof is interesting but hard.

# Intermediary Sets

Are there sets that are both

1. Not decidable
2. Weaker than HALT.

Answer: YES and the proof is interesting but hard.
**Verdict** A step backwards but a very interesting proof.

# More Kolmogorov

I could apply Kolm Complexity to

# More Kolmogorov

I could apply Kolm Complexity to

1. Proving more langs not regular.

# More Kolmogorov

I could apply Kolm Complexity to

1. Proving more langs not regular.
2. Proving some langs have large DFAs, NFAs, CFGs.

# More Kolmogorov

I could apply Kolm Complexity to

1. Proving more langs not regular.
2. Proving some langs have large DFAs, NFAs, CFGs.
3. Getting Avg Case Analysis of some algorithms.

# Misc

Exposition by William Gasarch—U of MD

# Parallel, Randomized, Quantum

There are other modes of computation.

# Parallel, Randomized, Quantum

There are other modes of computation.

1. **Parallelism** There is a theory analogous to P vs NP to show problems can't be parallelized.

# Parallel, Randomized, Quantum

There are other modes of computation.

1. **Parallelism** There is a theory analogous to P vs NP to show problems can't be parallelized.
2. **Randomized Computations** How much does randomization help?

# Parallel, Randomized, Quantum

There are other modes of computation.

1. **Parallelism** There is a theory analogous to P vs NP to show problems can't be parallelized.
2. **Randomized Computations** How much does randomization help?
3. **Quantum Computing** there is a notion of quantum-DFA that I could look into and do, but might be too hard. For me!

# Parallel, Randomized, Quantum

There are other modes of computation.

1. **Parallelism** There is a theory analogous to P vs NP to show problems can't be parallelized.
2. **Randomized Computations** How much does randomization help?
3. **Quantum Computing** there is a notion of quantum-DFA that I could look into and do, but might be too hard. For me!

**Verdict** I would have to look into all of these more to see if they make sense. Quantum would be a step forward.

# Complexity Classes Based on Problems

# Complexity Classes Based on Problems

1. All-pairs-shortest-path **seems to require** $\Omega(n^2)$ time. There are now APSP-hard problems.

# Complexity Classes Based on Problems

1. All-pairs-shortest-path **seems to require** $\Omega(n^2)$ time. There are now APSP-hard problems.
2. There are others problems that are thought to be hard that are used to show that other problems are thought to be hard.

# What to take Out (Brief)

Exposition by William Gasarch—U of MD

# What to Take Out?

If I want to put any of that in, I need to take some stuff out.

# What to Take Out?

If I want to put any of that in, I need to take some stuff out.

1. CSL's I could easily take out. :-)

# What to Take Out?

If I want to put any of that in, I need to take some stuff out.

1. CSL's I could easily take out. :-)
2. Decidability of $(\mathbb{Q}, <)$ can go.

# What to Take Out?

If I want to put any of that in, I need to take some stuff out.

1. CSL's I could easily take out. :-)
2. Decidability of $(\mathbb{Q}, <)$ can go.
3. Could reduce how much time I spend on regular by cutting out Regular Expressions. They are done in 330 anyway. DO want to keep the SMALL-NFA-RESPECT problem.

# BILL AND NATHAN RECORD LECTURE!!!!

BILL AND NATHAN RECORD LECTURE!!!