6.1400

Lecture 24:
Finish Randomized Complexity, Begin Review

Final Exam Information

Who: You
On What: Everything through BPP (today)
With What: One sheet (*double-sided*) of notes are allowed
When: Wed, May 21 **1:30PM - 4:30PM**
Where: **HERE, 37-212**
Why: Because you will ace it
How: By studying

Practice final exam coming out soon!

The Plan For The Next Two Lectures

Today: We'll finish randomized complexity,
and start reviewing the major topics

Thursday: When the review is done...
Ask Me Anything!

Ask questions in person, or post questions anonymously on piazza.
I will answer them in class on Tuesday!

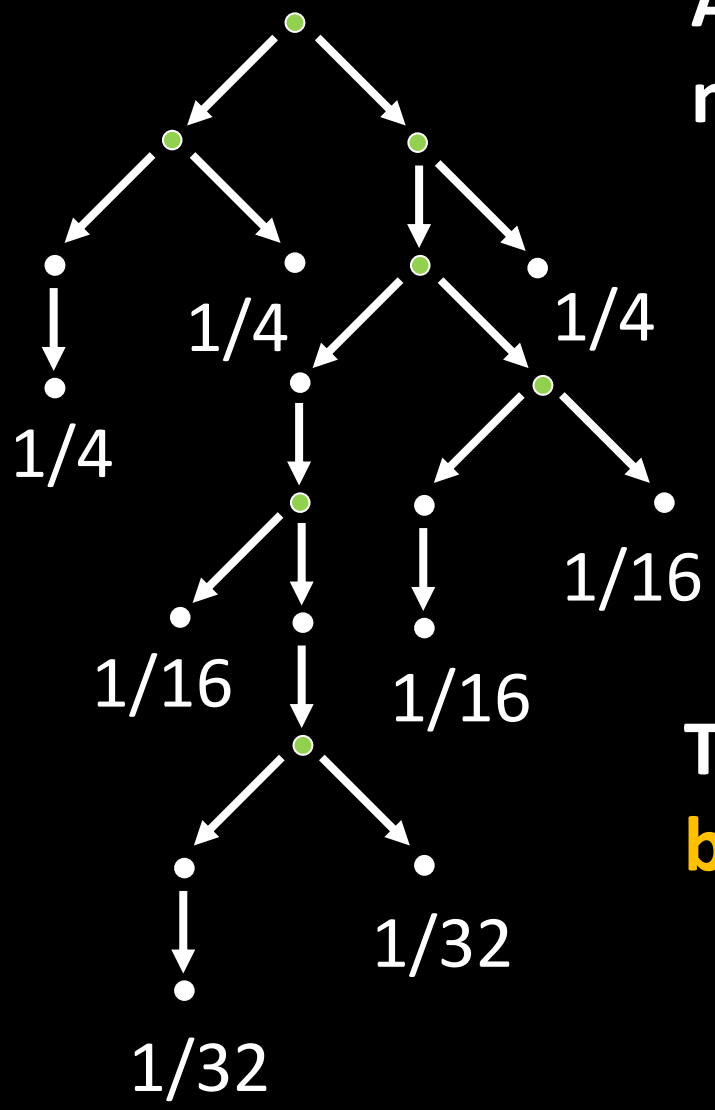
VOTE VOTE VOTE

For your *favorite* course on automata and complexity

Please complete the online subject evaluation for 6.1400

Randomized / Probabilistic Complexity

Probabilistic TMs



A **probabilistic TM M** is a nondeterministic TM where:

- Each nondeterministic step is called a **coin flip**
- Each nondeterministic step has only two legal next moves **(heads or tails)**

The probability that M runs on a **branch b** is: $\Pr[b] = 2^{-k}$

where k is the number of coin flips that occur on branch b

Definition. A probabilistic TM M decides a language A with error ε if for all strings w,

$w \in A \Rightarrow \Pr[\ M \text{ accepts } w\ ] \geq 1 - \varepsilon$

$w \notin A \Rightarrow \Pr[\ M \text{ doesn't accept } w\ ] \geq 1 - \varepsilon$

Theorem: A language A is in NP if there is a nondeterministic polynomial time TM M such that for all strings w:

$w \in A \Rightarrow \Pr[\ M \text{ accepts } w\ ] > 0$

$w \notin A \Rightarrow \Pr[\ M \text{ accepts } w\ ] = 0$

BPP = Bounded Probabilistic P

BPP = { L | L is recognized by a probabilistic polynomial-time TM with error at most 1/3 }

Why 1/3?

It doesn't matter what error value we pick, as long as the error is smaller than 1/2.

When the error is smaller than 1/2, we can make it very small by repeatedly running the TM.

An arithmetic formula is like a Boolean formula, except it has +, –, and * instead of OR, NOT, AND.

ZERO-POLY = { p | p is an arithmetic formula that is *identically* zero}

Identically zero means: all coefficients are 0

Theorem: ZERO-POLY $\in$ BPP

Big Idea: Evaluate p on *random values*

Theorem (Schwartz-Zippel-DeMillo-Lipton)

Let $p(x_1,x_2,\ldots,x_n)$ be a *nonzero* polynomial, where each x_i has degree at most d. Let $F \subset Z$ be finite.

If $a_1,\ldots, a_n$ are selected randomly from F, then:

$$\Pr [\ p(a_1, \ldots, a_n) = 0\ ] \leq dn/|F|$$

ZERO-POLY = { p | p is an arithmetic formula that is *identically* zero}

Theorem: ZERO-POLY $\in$ BPP

Proof: Suppose $n=|p|$. Then p has $k \le n$ variables, and the *degree* of each variable is at most n.

Algorithm A: Given arithmetic formula p,
For all $i = 1,\ldots,k$, choose r_i randomly from $\{1,\ldots,3n^2\}$
If $p(r_1, \ldots, r_k) = 0$ then output *zero*
else output *nonzero*

Observe A runs in polynomial time.

If $p \equiv 0$, then Pr[A(p) outputs *zero*] = 1

If $p \not\equiv 0$, then by the Schwartz-Zippel lemma,

Pr[A(p) outputs *zero*] = $\Pr_r[p(r) = 0] \le n^2/3n^2 \le 1/3$

Checking Equivalence of Arithmetic Formulas

ZERO-POLY = { p | p is an arithmetic formula that is identically zero}

Theorem: ZERO-POLY $\in$ BPP

EQUIV-POLY = { (p,q) | p and q are arithmetic formulas computing the same polynomial}

Corollary: EQUIV-POLY $\in$ BPP

Proof: (p,q) in EQUIV-POLY $\Leftrightarrow$ p-q in ZERO-POLY

Therefore EQUIV-POLY $\leq_p$ ZERO-POLY

and we get a BPP algorithm for EQUIV-POLY.

See Sipser 10.2 for an application to testing equivalence of simple programs!

Equivalence of Arithmetic Formulas

EQUIV-POLY = { (p,q) | p and q are arithmetic formulas computing the same polynomial}

Corollary: EQUIV-POLY $\in$ BPP

There is a big contrast with Boolean formulas!

EQUIV = { (ϕ,ψ) | ϕ and ψ are Boolean formulas computing the same function}

We showed EQUIV is in coNP. It's also coNP-complete!
TAUTOLOGY $\leq_P$ EQUIV: map ϕ to (ϕ, True)

ZERO-POLY = { p | p is an arithmetic formula that is identically zero}

Theorem: ZERO-POLY $\in$ BPP

It is not known how to solve ZERO-POLY efficiently *without* randomness!

Thm [KI'04, AvM'11] IF ZERO-POLY $\in$ P
THEN NEW *LOWER BOUNDS FOLLOW*
(not P ≠ NP, but still a breakthrough!)

BPP = { L | L is recognized by a probabilistic polynomial-time TM with error at most **1/3** }

Is BPP $\subseteq$ NP?

Is BPP $\subseteq$ PSPACE?

Is NP $\subseteq$ BPP?

Is BPP = EXPTIME?

Definition: A language A is in **RP (Randomized P)** if there is a nondeterministic polynomial time TM M such that for all strings x:

$$x \notin A \Rightarrow \Pr[M(x) \text{ accepts}] = 0$$

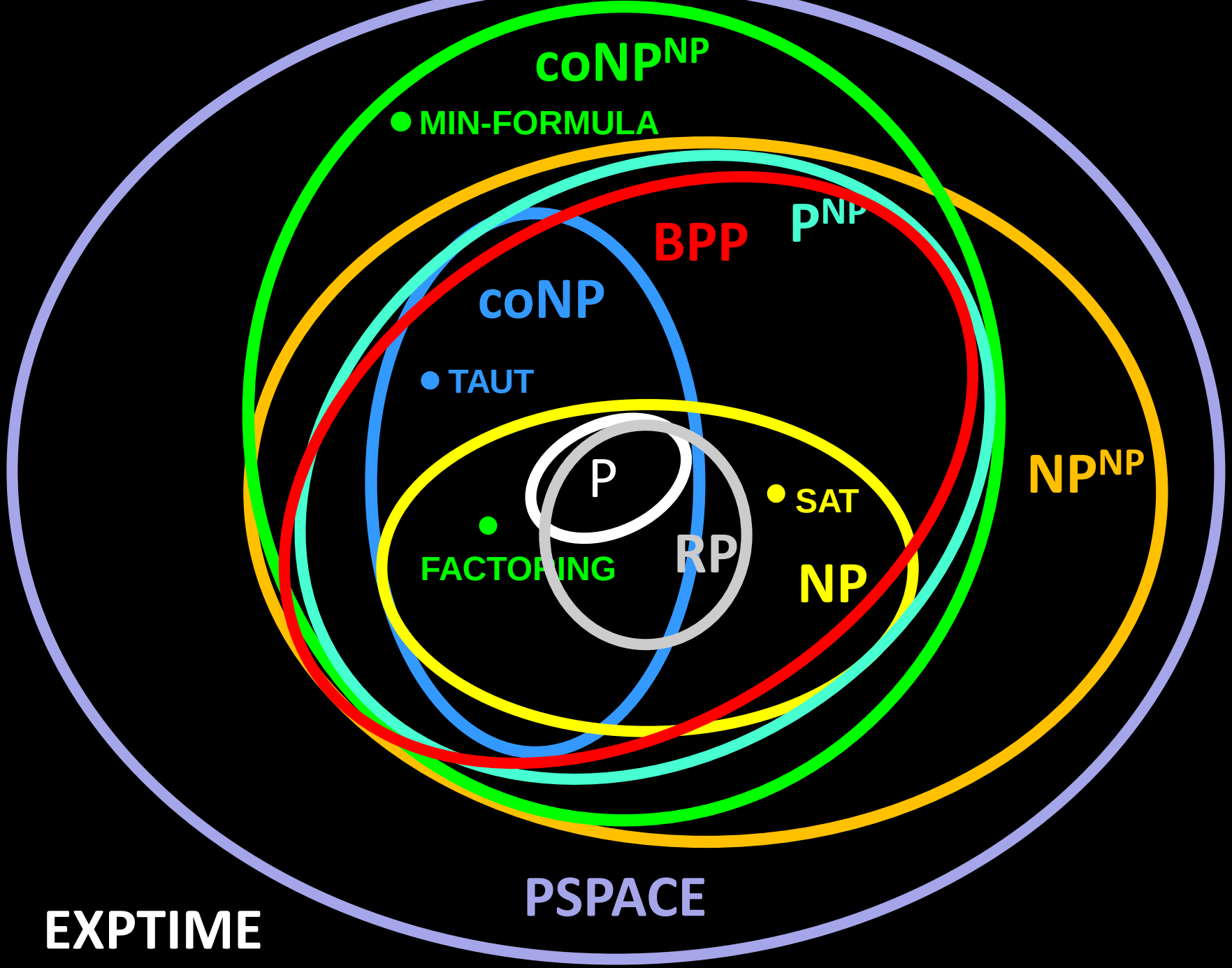
$$x \in A \Rightarrow \Pr[M(x) \text{ accepts}] > 2/3$$

NONZERO-POLY = { p | p is an arithmetic formula that is not identically zero}

Theorem: NONZERO-POLY $\in$ RP
(Our proof of ZERO-POLY in BPP shows this)

Is RP $\subseteq$ NP?

Is RP $\subseteq$ BPP?


coNP^NP
MIN-FORMULA
BPP
P^NP
coNP
TAUT
P
SAT
FACTORING
RP
NP
NP^NP
PSPACE
EXPTIME