Today:

- Oracles
- Randomized Complexity Classes

Recall, P^{SAT} is the class of poly-time algorithms with a SAT oracle. $NP \subseteq P^{SAT}$, and $coNP \subseteq P^{SAT}$

Similarly NP^{SAT} is the class of nondeterministic poly-time algorithms with a SAT oracle

Q| Let NON-MIN-FORMULA = $\{\langle\phi\rangle \mid \phi$ has a smaller, equivalent formula$\}$

Show NON-MIN-FORMULA $\in NP^{SAT}$

A| "On input w

1) Nondeterministically guess a smaller formula ϕ'
2) Decide if ϕ' is equivalent to ϕ. If so, accept. "

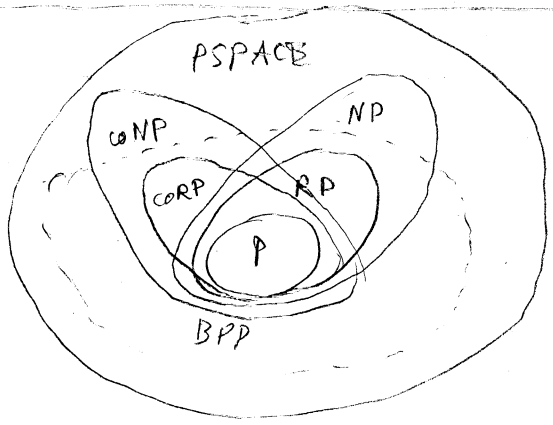
↑

Must do step 2 deterministically. How?

Note EQUIVALENT = $\{\langle\phi,\phi'\rangle \mid \phi$ and ϕ' are equivalent$\} \in NP$

so EQUIVALENT $\in coNP$

so EQUIVALENT $\in P^{SAT}$



Review

BPP = Prob Poly Time w/ 2-sided error

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

$w \notin A \Rightarrow Pr[M \text{ accepts } w] < 1/3$

RP = Prob Poly Time w/ 1-sided error ("too critical")

$w \in A \Rightarrow Pr[M \text{ accepts } w] > 1/2$

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

coRP = complement of RP ("too lenient")

Facts: $P \subseteq RP \overset{\text{why}}{\subseteq} NP \subseteq PSPACE$

$\subseteq coRP \subseteq coNP \subseteq$

$RP \subseteq$

$coRP \subseteq BPP \subseteq PSPACE$

↑ we will show this

Open questions: $BPP \subseteq NP$?

$NP \subseteq BPP$?

Interesting Theorem (don't need to know): $BPP \subseteq NP^{SAT}$

Also interesting: these days, most experts think that $P = BPP$ (but nobody knows)

Note this implies that if $P = NP$, then $P = BPP$

Q | Show $BPP \subseteq PSPACE$

Proof | Suppose M is a BPP machine, runs in time $O(n^k)$

Then $\exists M'$, also in BPP, which decides the same language and makes exactly cn^k coin tosses on every run

Design deterministic D, deciding the same language

D = " On input w

1) For each cn^k bit binary string r
   - Run M on w, using r as the sequence of coin flips
   - Increment counter if M accepts

2) If counter $> \frac{2}{3}(2^{|r|})$, accept

   If counter $< \frac{1}{3}(2^{|r|})$, reject "

D uses poly space for storing r and the counter, and simulating poly-time M

Amplification Lemma:

Let M be a BPP machine w/ $Pr[\text{error}] \leq \frac{1}{3}$

Then $\exists$ BPP machine M' w/ $Pr[\text{error}] \leq \frac{1}{poly(n)}$ for any $poly(n)$

M' = " on input w

1) Run M on w $2K$ times

2) Take majority vote "

What should K be? (see book for easy calculation)