Today’s Topics

- Correction: NOT-STRONGLY-CONNECTED ∈ NL
- Prove: $\text{EQ}_{\text{REX}} \in \text{PSPACE}$
- $P^{\text{TQBF}} = \text{NP}^{\text{TQBF}}$
- $P^A \neq \text{NP}^A$
- Prove: MIN-FORMULA ∈ coNP$^{\text{SAT}}$
- Review: BPP
- $P \subseteq \text{BPP}$, BPP ⊆ PSPACE
Correction: NOT-STRONGLY-CONNECTED ∈ NL

Recall: Trying to show STRONGLY-CONNECTED ∈ NL
(path exists from every node to every other node in directed graph)

Show: NOT-STRONGLY-CONNECTED ∈ NL = coNL

NOT-STRONGLY-CONNECTED = “On input G,

1. nondet. guess two vertices u,v
2. Return NOT-PATH(G, u, v)”

Note: NOT-PATH is in coNL = NL, so can invoke it in NL TM.
Prove: $\text{EQ}_{\text{REX}} \in \text{PSPACE}$

Definition: $\text{EQ}_{\text{REX}} = \{ <R_1, R_2> \mid \text{where } R_1 \text{ and } R_2 \text{ are equivalent reg. exprs} \}$

Proof: Show $\neg \text{EQ}_{\text{REX}} \in \text{NPSPACE} = \text{PSPACE} \rightarrow$ we can negate the result

Let $M =$ “On input $<R_1, R_2>$

1. Convert $R_1$ and $R_2$ to equivalent NFAs $N_1$ and $N_2$ having $m_1$ and $m_2$ states
2. Nondet. guess the symbols one-by-one of a string $s$ of length $2^{m_1 + m_2}$ and simulate $N_1$ and $N_2$ on $s$, storing only the current sets of states of $N_1$ and $N_2$
3. If they ever disagree on acceptance, then accept
4. If they always agree on acceptance then reject”
\( \mathbf{P}^{\text{TQBF}} = \mathbf{NP}^{\text{TQBF}} \)

Statement: \( \mathbf{NP}^{\text{TQBF}} \subseteq \mathbf{NPSPACE} = \mathbf{PSPACE} \subseteq \mathbf{P}^{\text{TQBF}} \)

First: \( \mathbf{NP}^{\text{TQBF}} \subseteq \mathbf{NPSPACE} \)

- Any time TQBF oracle is invoked, NPSPACE TM can simply compute that result

Second: \( \mathbf{NPSPACE} = \mathbf{PSPACE} \) \quad \text{Savitch's Theorem}

Third: \( \mathbf{PSPACE} \subseteq \mathbf{P}^{\text{TQBF}} \)

- Reduce any PSPACE language to TQBF and ask the oracle
$P^A \neq NP^A$

Idea: Force a search of the oracle’s language that is proveably not polynomial

For oracle $A$, define $L = \{ \text{strings } w \mid \exists x \in A \text{ s.t. } |x| = |w| \}$

Note: $L \in NP^A$

Construct $A$ such that $L \notin PA$

The oracle $A$ does not return the $x$ that works within polynomial amount of steps. This force of search is means that $L$ must return a result after a polynomial number of steps.

If $L$ accepts, $A$ shall never include $x$. If $L$ rejects, then in some exponential number of steps in the future, it should return $x$. Thus $L$ cannot determine if $w$ is in the language correctly in a polynomial number of steps.
Implications: $P^{\text{QBF}} = NP^{\text{QBF}}$ but $P^A \neq NP^A$

If $P = NP$ were to be proven by some procedural construction such as Savitch’s Theorem showed $\text{PSPACE} = \text{NPSPACE}$

→ Then for every oracle $X$ applied, $P^X = NP^X$

However, showed that an oracle $A$ exists such that $P^A \neq NP^A$

- This means cannot show $P=NP$ via a direct construction.
- Would need to prove via non-relavitizable methods such as arithmetization.

Currently, expectation is overwhelmingly $P \neq NP$ for this reason.
Prove: MIN-FORMULA $\in$ coNP$^{\text{SAT}}$

Definition: $\text{EQ}_{BF} = \{<\varphi_1, \varphi_2> \mid \varphi_1 \text{ and } \varphi_2 \text{ are equivalent boolean formulas}\}$
$\text{EQ}_{BF} \in \text{coNP}$ because $\neg \text{EQ}_{BF} \in \text{NP}$ simply

Proof for $\neg \text{MIN-FORMULA} \in \text{NP}^{\text{SAT}}$

M = “On input $<\varphi>$

1. Nondet. guess boolean formula $\varphi'$ that is shorter than $\varphi$
2. Ask SAT oracle if $<\varphi, \varphi'> \in \neg \text{EQ}_{BF}$ (reduce $\neg \text{EQ}_{BF}$ problem to SAT problem)
3. If oracle answers “no”, namely that $\varphi$ and $\varphi'$ are equivalent, so accept
4. Otherwise, reject”
Review: BPP

BPP = \{ A \mid \text{exists a poly-time Probabilistic TM that decides } A \text{ with error } \epsilon = 1/3 \}

or \( \epsilon < 1/2 \)

Amplification Lemma: If \( M_1 \) is a poly-time PTM with error \( \epsilon_1 = 1/3 \) then,
for any \( 0 < \epsilon_2 < 1/2 \), there is an equivalent poly-time PTM \( M_2 \) with error \( \epsilon_2 \)
Can strengthen to make \( \epsilon_2 < 2^{-1} \cdot \text{poly(n)} \)

Run \( M_1 \) k times and return majority result which reduces error probability

Significance: Can make the error probability arbitrarily small (never 0 however!)
Review: BPP

computation tree for $M$ on $w$
Review: BPP

NP

$w \in A$

$\geq 1$ accepting

Computation trees for $M$ on $w$

$w \notin A$

all rejecting

BPP

Many accepting

Few rejecting

Few accepting

Many rejecting
\( P \subseteq \text{BPP}, \text{BPP} \subseteq \text{PSPACE} \)

\( P \subseteq \text{BPP} \)

- Statement: a BPP TM can decide all languages in \( P \)

\( \text{BPP} \subseteq \text{PSPACE} \)

- Statement: a PSPACE TM can decide all languages in BPP