Lecture 16
CSE 431
Intro to Theory of Computation

Last time:
RELPRIME EP

\[ \text{FACTOR} = \{ <N, k, l> : \text{integer } N \text{ has a factor } n \text{ with } k \leq m \leq l \} \]

FACTOR \in NP \quad \text{EP?} \quad \text{open}

Textbook: COMPOSITES = \{ <N> : N \text{ is an integer s.t. } \exists p, q \text{ s.t. } pq = N \text{, } 1 < p, q < N \} \]

\( \langle N \rangle \in \text{COMPOSITES} \iff \langle N, 2, N-1 \rangle \in \text{FACTOR} \)

\forall A \in \text{CFL}, A \in \text{EP}

Alternate characteristics of NP

Recall that
\[ \text{NP} = \bigcup_{k} \text{NTIME}(n^k) \]
set of languages

We will use notation \( N^{\text{polynomial}} \) to mean \( N^k \) for some \( k \) "polynomial"
Thm 1. A \in NP

\[ L \implies 2 \]\text{Deterministic polytime TM } V\text{ s.t.} \forall x\in A \implies \exists y \text{ s.t. } \|y\| = \|x\|_1 \text{ and } V \text{ accepts } (x,y).

\[ \implies 3 \]text
\[ \exists \text{ deterministic TM } V' \text{ s.t. } V' \text{ on input } (x,y) \text{ runs in time polynomial in } \|x\| \text{ and } \forall x\in A \implies \exists c \text{ s.t. } V' \text{ accepts } (x,c) \text{ (certificate or proof that } x \in L) \]

\[ \text{Example using det TM: } \]
\text{COMPOSITES } \in \text{ NP:}
\begin{align*}
\text{For input } N & \text{ (a bit)} \\
\text{Certificate: } & \text{ integers } p,q \text{ (each at most } b \text{ bit)} \\
\text{Verify: } & \text{ check } N = pq \text{ (time at most } O(n^2))
\end{align*}

\text{Proof of Thm 1.}

First observe that \[ 2 \iff 3 \]

\[ 1 \implies 3 \] : We can use \( c = y \) and \( V' = V \)
\[ \text{only } \|y\| = \|x\|_1 \text{ (total running time of } V' \text{ is } 1\times|0||1|) \text{ as required} \]

\[ 3 \implies 2 \] : Suppose there is such a \( V' \). We use \( V = V' \). Since \( V' \) runs in \( 1\times|0||1| \text{ time, it only can look at first } 1\times|0||1| \text{ bits of } c \) \text{, let } y \text{ be those bits.}
$1 \Rightarrow 0$: Given vertices $V$ for $A$; create NTM for $A$ as follows:

- On input $X$:
  - Use nondeterministic guesses to guess a string $y$ of length $|X|/2n$ and create string $<X,y>$.
  - Run $V$ on input $<X,y>$. accept if $V$ does.

Total time is polynomial.

$0 \Rightarrow 1$:
Suppose we have an NTM $M$ for $A$; in general $M$ will have a computation tree that branches at any time.

Recall simulation of NTMs by TMs: (BFS of tree)

- Input
- Address tape
- Simulation tape

Steps:
- Copy input to tape 3
- Use address tape on tape 2 to deterministically simulate on tape 3, if $M$ accepts then accept;
- Increment address 2 and repeat.

Idea here: certificate $y$ is a possible address
- of polynomial length
- Poly time since
- # of steps = poly of $|y|$.

Hypothesis $V$ checks that simulation on input $X$ with address string $y$ accepts.
More examples in NPs:

**HAMPATH = \{ G \}**: G is a directed graph with a path that touches each vertex exactly once.

**HAMPATH NP:**
- *Input* \( G \):
  - \( n \) = \# vertices of \( G \)
- *Certificate*:
  - A sequence of vertices of length \( n \)
- *Verify*:
  - All vertices are different
  - Each adjacent pair of vertices \( u, v \) has a directed edge \( (u, v) \) in \( G \)

**SATE \{ \Phi \}**: \( \Phi \) is a Boolean (propositional logic) formula that is satisfiable, has a truth assignment that makes it true.

**SAT NP**:
- *Polynomial*
  - *Certificate*:
    - Truth assignment \( \alpha \)
  - *Verify*:
    - \( \Phi \) evaluates to true under assignment \( \alpha \)
- \( \Phi \) is a Boolean formula

**CNFSAT = \{ \Phi \}**: \( \Phi \) is a satisfiable formula in Conjunctive Normal Form (CNF).

Recall CNF from 310: (aka. "product-of-sums")
Recall: A CNF is an $\land$ of clauses, a clause is an $\lor$ of literals, a literal is a propositional logic variable or its negation, $x_i$ or $\overline{x_i}$.

CNFSAT in NP: same certificate as SAT
- only change is that verifier checks that $\Psi$ is a CNF, i.e. add rule to checking $\Psi$.

Def: $\text{RCNF formula}$ is a CNF formula where each clause has length $\leq k$ ($\leq k$ literals)

RSAT = $\{<\Psi>: \Psi$ is a satisfiable RCNF$\}$

RSAT in NP: as before but check RCNF format of input, too.

Relation among problems:

Def: Given $A, B \in \Sigma^*$, $A$ is polynomial-time mapping reducible to $B$, written $A \leq_p B$, or $A \leq_m B$ (most standard)

iff there is a poly-time computable function $f: \Sigma^* \rightarrow \Sigma^*$

only differ from $A \leq m B$

$\forall x \in \Sigma^*, \exists y \in A \Rightarrow f(x) \in B$
Then, if $A \leq^P B$ and $B \in P$ then $A \in P$.

If $A \leq^P B$ and $B \in NP$ then $A \in NP$.

Proof. As usual, given polytime machine $M_B$ for $B$ and $M_f$ for reduction $f$ define

$$M_A(x) = \begin{cases} 1 & f(x) \in \text{accept} \\ 0 & f(x) \in \text{reject} \end{cases}$$

This is correct as always. Only need to check run time.

Since both are polytime, let runtime of $M_f$ be $O(n^k)$.

Thus runtime of $M_B$ be $O(n^k)$.

Total runtime for $M_A$ on input $x$

$$O(1|x|^k) + O(1|x|^l)$$

which is $O(1|x|^k) + O\left(\frac{O(1|x|^k)}{l}\right)$ which is polynomial in size of output at most runtime.

For NTM run argument applies.