1 Two Equivalent Definitions of NP

Recall: Non-deterministic Turning Machines (NDTMs)

The configuration of a Turing Machine (TM) includes the current state, contents of non-blank work tape cells, and head locations.

In the following diagram, each circle represents a configuration of a NDTM:

The transition functions are $\delta_i : Q \times \Gamma^{k+2} \rightarrow Q \times \Gamma^{k+1} \times \{L, R, S\}^{k+2}$. $x$ is accepted by the NDTM $N$ if there exists a sequence of $\delta_0$ and $\delta_1$'s such that $N$ halts and writes 1 on the output tape when it follows that sequence.

We say that the NDTM runs in time $T(n)$ if the depth of the computation tree is $T(n)$. Note that the number of possible paths is exponential in $T(n)$.

**Definition 1.1.** $L \in \text{NP}$ if there exists a polynomial time verifier $V$ such that for all $x \in \{0, 1\}^n$, $x \in L$ if and only if there exists some $y \in \{0, 1\}^{c|x|c}$ such that $V(x, y) = 1$.

**Definition 1.2.** $L \in \text{NP}$ if there exists a polynomial time NDTM $N$ that computes $L$.

**Claim 1.3.** The two definitions of NP are equivalent.
Proof. First suppose that \( L \in \text{NP} \) in the sense of Definition 1.1. Then we have a verifier \( V \) such that \( x \in L \iff \exists y \in \{0,1\}^{c|x|^c}, V(x,y) = 1 \). The way we construct \( N \) is by “non-deterministically guessing \( y \) and simulating \( V(x,y) \)”. More specifically, consider the following NDTM:

\[
N \quad \text{guess tape}
\]

After computing \( |x| \), the length of the input, the NDTM writes a guess of \( y \) on the guess tape in the first \( c|x|^c \) steps. Then, it simulates \( V \) on \( (x,y) \) and outputs according to \( V \). The correctness and the runtime of the NDTM are straightforward.

Next, suppose that \( L \in \text{NP} \) in the sense of Definition 1.2. Then by assumption, we have a NDTM \( N \) that runs in \( cn^c \) time and computes \( L \). The verifier \( V \) is constructed as follows: On input \( x \) and certificate \( y \in \{0,1\}^{c|x|^c} \), simulate \( N \) on \( x \) using the \( i \)th bit of \( y \) to choose \( \delta_0 \) or \( \delta_1 \).

\[
\square
\]

2 Reductions and NP-completeness

**Definition 2.1.** We say that \( L_1 \) is poly-time Karp reducible to \( L_2 \) \( (L_1 \leq_P L_2) \) if there exists a polynomial time TM \( M \) such that \( x \in L_1 \iff M(x) \in L_2 \).

**Definition 2.2.** \( L \) is **NP-complete** if \( L \in \text{NP} \) and for all \( L' \in \text{NP}, L' \leq_P L \).

**Example 2.3.** Some examples of NP-complete problems: SAT, 3SAT (Cook-Levin), 3-Coloring, TSP.

Many computer scientists believe that \( P \neq \text{NP} \):

\[
\text{NP} \quad \text{NP-complete}
\]

We’ll prove in the next lecture that if \( P \neq \text{NP} \), then there exists a problem in \( \text{NP} \) that is not in \( P \) and is not NP-complete.
3 Time Hierarchy Theorem

Definition 3.1. $T(n)$ is time-constructible if a TM on input $1^n$ can write $1^{T(n)}$ in $cT(n)$ time and $T(n) \geq n$.

Example 3.2. Examples of time-constructible functions include $2^n, 2^{2^n}$, and $2^{2\sqrt{\log n}}$.

Exercise 3.3. Construct a function $T$ satisfying $T(n) \geq n$ that is not time-constructible.

Theorem 3.4. Suppose $f$ and $g$ are time-constructible functions from $\mathbb{N}$ to $\mathbb{N}$, and $f(n) \cdot \log f(n) = o(g(n))$. Then $\text{DTIME}(f(n)) \subsetneq \text{DTIME}(g(n))$.

Proof. Consider the following TM $D$: On input $w = (x, y)$, simulate $M_x$ on $w$ for $g(|w|)$ simulation steps. If $M_x$ halts at some point, flip the output. Otherwise, output 0.

By construction, $L(D) \in \text{DTIME}(g(n))$. Now let $x^*$ be such that $M_{x^*}$ runs in $cf(n)$ time. Then we claim that $L(M_{x^*}) \neq L(D)$. In other words, there exists a $w$ such that $M_{x^*}(w) \neq D(w)$.

To prove the claim, notice that $M_{x^*}$ on input $w = (x^*, 1^k)$ can be simulated in $C' \cdot f(|w|) \cdot \log(f(|w|))$ steps. This is less than $g(|w|)$ for large enough $k$. Therefore, $M_{x^*}$ halts on $w$ in less than $g(|w|)$ steps, so by the definition of $D$, $M_{x^*}(w) \neq D(w)$. This shows that $\text{DTIME}(f(n)) \subsetneq \text{DTIME}(g(n))$. \qed

Remark 3.5. Using $1^k$ as a part of the input is called the “padding argument”. It is useful here because the description of the TM $M_{x^*}$ may not be long enough for the asymptotic bound $f(n) \cdot \log f(n) = o(g(n))$ to apply.

Remark 3.6. This argument does not generalize directly to NDTMs because we can’t flip the output of NDTMs.