Computer Science 511

Introduction to Nondeterministic Computation

Instructor: Wayne Eberly

Department of Computer Science
University of Calgary

Lecture #8

Goals for Today

- Introduce and relate two notions of “nondeterministic computation”
- Introduce the complexity classes NTIME(f) and $\mathcal{NP}$, and relate these to deterministic complexity classes

Nondeterministic Turing Machines

Definition: A k-tape ***nondeterministic Turing machine*** is a machine

$$M = (Q, \Sigma, \Gamma, \delta, q_0, q_{\text{accept}}, q_{\text{reject}})$$

such that

- Q, Σ, Γ, q_0 q_{accept} and q_{reject} are as defined for (standard) *deterministic* Turing machines;
- The transition function, δ, is now a total function

$$\delta : Q \times \Gamma^k \to \mathcal{P}(Q \times (\Gamma \times \{\mathtt{L}, \mathtt{R}, \mathtt{S}\})^k)$$

  where the "power set" $\mathcal{P}(S)$ of any finite set S is the set of all *subsets* of S.

Nondeterministic Turing Machines

- Since q_{accept} and q_{reject} are "halting states,"

$$\delta(q_{\text{accept}}, \sigma_1, \sigma_2, \ldots, \sigma_k) = \delta(q_{\text{reject}}, \sigma_1, \sigma_2, \ldots, \sigma_k) = \emptyset$$

for all symbols $\sigma_1, \sigma_2, \ldots, \sigma_k \in \Gamma$.

Nondeterministic Turing Machines

- Computation of a nondeterministic Turing machine M on a string $\omega \in \Sigma^\star$ can be modelled as a ***tree*** with ***configurations*** of the machine at its nodes.
  - The initial configuration of M on input ω is the configuration at the ***root*** of this tree.
  - If a configuration C of M has the machine in state q, with symbols $\sigma_1, \sigma_2, \ldots, \sigma_k$ visible on its tapes, then the number of children of the node with this configuration is equal to the size of the set $\delta(q, \sigma_1, \sigma_2, \ldots, \sigma_k)$ — and there is node for the configuration obtained by applying each of the transitions in this set.
  - Thus the node for a configuration is a leaf in this tree if and only if this is a halting configuration.

Nondeterministic Turing Machines

- Thus each ***path down*** through this tree (starting from the root) corresponds to one possible computation of M on its input string — corresponding to a series of ***guesses*** about which possible transition to apply.
- M ***accepts*** a string ω if and only if there exists at least one path leading to a configuration with M in its accepting state q_{accept}.
- M ***recognizes*** a language $L \subseteq \Sigma^\star$ if L is the set of strings in $\Sigma^\star$ that M accepts (as defined above).
- M ***decides*** a language $L \subseteq \Sigma^\star$ if
  - M recognizes L and, furthermore
  - The tree of configurations for $\mathcal{M}$ on input ω is ***finite***, for ***every*** string $\omega \in \Sigma^\star$.

Nondeterministic Turing Machines

- The ***time*** used by M on input $\omega \in \Sigma^\star$ is defined to be the ***depth*** of the tree of configurations for M on input ω — that is, the ***maximum*** of the length of any path from the root down to any leaf in this tree.
- If $f : \mathbb{N} \to \mathbb{N}$ then M ***decides*** L ***in time*** f if M decides L, and the time used by M on input ω is at most $f(|\omega|)$ for *every* string $\omega \in \Sigma^\star$.
- *One way* to define NTIME(f) is as follows: NTIME(f) is the set of languages $L \subseteq \Sigma^\star$ (for some alphabet Σ) such that there exists a nondeterministic Turing machine M that decides L using time in $O(f)$.

  Older textbooks use this definition; most recent ones use a different definition, as described next.

Verification of a Language

- Once again, consider a language $L \subseteq \Sigma^\star$. Let Σ_C be another (possibly different) alphabet, suppose that # is a symbol such that $\# \notin \Sigma \cup \Sigma_C$, and let

$$\widehat{\Sigma} = \Sigma \cup \Sigma_C \cup \{\#\}.$$

Then an *ordered pair*, consisting of a string $\omega \in \Sigma^\star$ and a string $\mu \in \Sigma_C^\star$ can be represented using the string

$$\omega\#\mu \in \widehat{\Sigma}^\star$$

— which includes exactly one #.

Verification of a Language

Definition: A ***verifier*** for a language L is a ***deterministic*** Turing machine M — with the following properties.

- The input alphabet for M is an alphabet $\widehat{\Sigma}$ as defined on the previous slide.
- M decides a language $\widehat{L} \subseteq \widehat{\Sigma}^{\star}$ that is a *subset* of the set

$$\{\omega\#\mu \mid \omega \in \Sigma^{\star} \text{ and } \mu \in \Sigma_C^{\star}\}.$$

- For every string $\omega \in \Sigma^{\star}$, $\omega \in L$ ***if and only if*** there exists ***at least one*** string $\mu \in \Sigma_C^{\star}$ such that $\omega\#\mu \in \widehat{L}$.

Σ_C is called the ***certificate alphabet*** for the verifier M. If $\mu \in \Sigma_C^{\star}$ such that $\omega\#\mu \in \widehat{L}$ then μ is a ***certificate*** for ω.

Verification of a Language: Another Definition of NTIME

Definition: Let $f : \mathbb{N} \to \mathbb{N}$. A language $L \subseteq \Sigma^{\star}$ is in $\mathrm{NTIME_V}(f)$ if there exists a verifier M (with certificate alphabet Σ_C) such that the number of steps used by M on any string $\omega\#\mu$, such that $\omega \in \Sigma^{\star}$ and $\mu \in \Sigma_C^{\star}$, is in $O(f|\omega|)$.

Note:

- The bound on the number of steps used by M, given above, depends on the length of the string $\omega \in \Sigma^{\star}$, and *not* on the rest of the input supplied to the verifier M. In particular, if *does not* depend on the length of μ.
- We do not worry about (or constrain) the number of steps used by M when its input *does not* have the form $\omega\#\mu$ for some string $\omega \in \Sigma^{\star}$ and $\mu \in \Sigma_C^{\star}$.

Equivalence of Definitions — Under a Reasonable Extra Condition

Recall that a function $f : \mathbb{N} \to \mathbb{N}$ is ***time-constructible*** if there is a deterministic Turing machine that maps the string 1^n to the binary representation of $f(n)$ using time in $O(f(n))$.

Claim #1: Let $f : \mathbb{N} \to \mathbb{N}$ be a time-constructible function such that $f(n) \in \Omega(n)$. Then $\mathsf{NTIME}(f) = \mathsf{NTIME}_\mathsf{V}(f)$.

The proof has two components:

(a) To prove that $\mathsf{NTIME}(f) \subseteq \mathsf{NTIME}_\mathsf{V}(f)$, choose an arbitrary language $L \subseteq \Sigma^\star$ such that $L \in \mathsf{NTIME}(f)$, and prove that $L \in \mathsf{NTIME}_\mathsf{V}(f)$.

(b) To prove that $\mathsf{NTIME}_\mathsf{V}(f) \subseteq \mathsf{NTIME}(f)$, choose an arbitrary language $L \subseteq \Sigma^\star$ such that $L \in \mathsf{NTIME}_\mathsf{V}(f)$ and prove that $L \in \mathsf{NTIME}(f)$.

Equivalence of Definitions — Under a Reasonable Condition

Sketch of Proof — First Direction

Let $L \subseteq \Sigma^\star$ such that $L \in \mathrm{NTIME}(f)$.

- Then there exists a nondeterministic Turing machine

$$M = (Q, \Sigma, \Gamma, \delta, q_0, q_{\text{accept}}, q_{\text{reject}})$$

such that M decides L, and the depth of the computation tree for M and an input string $\omega \in \Sigma^\star$ is in depth $O(f(|\omega|))$. In particular, there exist non-negative integers c_0 and c_1 such that the depth of the computation tree is at most $c_1 f(|\omega|) + c_0$, for every string $\omega \in \Sigma^\star$.

Equivalence of Definitions — Under a Reasonable Condition

- Suppose M has k tapes. Since M is a fixed Turing machine there exists a positive constant $U \in \mathbb{N}$ such that

$$|\delta(q, \sigma_1, \sigma_2, \ldots, \sigma_k)| \leq U$$

  for all $q \in Q$ and $\sigma_1, \sigma_2, \ldots, \sigma_k \in \Gamma$.

- It is possible to *order* each of the transitions that can be chosen — that is, if $|\delta(q, \sigma_1, \sigma_2, \ldots, \sigma_k)| = m$, for some integer m such that $1 \leq m \leq U$, then one can treat this set as

$$\delta(q, \sigma_1, \sigma_2, \ldots, \sigma_k) = \{\chi_0, \chi_1, \chi_2, \ldots, \chi_{m-1}\}$$

  for $\chi_0, \chi_1, \chi_2, \ldots, \chi_{m-1} \in Q \times (\Gamma \times \{\mathtt{L}, \mathtt{R}, \mathtt{S}\})^k$, in such that a way that (for $0 \leq i \leq m-1$) one can determine χ_i, given i.

Equivalence of Definitions — Under a Reasonable Condition

- Now consider a "certificate alphabet"

$$\Sigma_C = \{\tau_0, \tau_1, \ldots, \tau_{U-1}\}$$

so that $|\Sigma_C| = U$.

- A ***verifier*** for L, with input alphabet $\widehat{\Sigma} = \Sigma \cup \Sigma_C \cup \{\#\}$, can be described that does the following.
  1. This ***rejects*** its input unless this has the form $\omega\#\nu$ where $\omega \in \Sigma^\star$, $\nu \in \Sigma_C^\star$, and the length of ν is at most $c_1 f(|\omega|) + c_0$.
  2. This uses the certificate ν, otherwise, to determine a specific ***path*** down the computation tree for M and ω — ***accepting*** ω if this path leads to an accepting configuration, and ***rejecting*** ω, otherwise.

Equivalence of Definitions — Under a Reasonable Condition

- Using the fact that f is time-constructible (which is important for the implementation and analysis of step 1), it can be shown that this verifies L using time in $O(f)$ — implying that $f \in \mathsf{NTIME}_{\mathsf{V}}(f)$.

Equivalence of Definitions — Under a Reasonable Condition

Sketch of Proof — Second Direction

Let $L \subseteq \Sigma^\star$ such that $L \in \mathrm{NTIME}_\mathrm{V}(f)$.

- Then there exist non-negative integers c_0 and c_1, and a verifier M (with some certificate alphabet Σ_C) that verifies L, using time at most $c_1 f(|\omega|) + c_0$ when the input is a string $\omega\#\nu$ for $\omega \in \Sigma^\star$ and $\nu \in \Sigma_C^\star$.
- A ***nondeterministic Turing machine*** $\widehat{M}$, with input alphabet Σ and a tape alphabet Γ with $\Sigma \cup \Sigma_C \cup \{\#\}$ as a subset, can decide L by carrying out the following, on input $\omega \in \Sigma^\star$.
  1. ***Guess*** a certificate $\nu \in \Sigma_C^\star$ with length at most $c_1 f(|\omega|) + c_0$, and append the string $\#\nu$ onto the input.
  2. Run M on the string that has been obtained — ***accepting*** if $\mathcal{M}$ accepts, and ***rejecting*** if $\mathcal{M}$ rejects.

Equivalence of Definitions — Under a Reasonable Condition

- The fact, that the function f is time-constructible, is needed to show that the first step can be carried out using $O(f(|\omega|))$ moves.
- Using this, it can be shown that the nondeterministic Turing machine $\widehat{M}$ has language L and that, for every input string $\omega \in \Sigma^\star$, the computation tree for $\widehat{M}$ and ω has depth in $O(f(|\omega|))$ — implying that $f \in \mathrm{NTIME}(f)$. □

The supplemental material lecture includes a more detailed proof of this result.

Relating Nondeterministic Computation to Deterministic Computation

Claim #2: TIME(f) $\subseteq$ NTIME(f) for every function $f : \mathbb{N} \to \mathbb{N}$.

Idea of Proof: Let $L \subseteq \Sigma^\star$ such that $L \in$ TIME(f).

- Then there exists a deterministic Turing machine M deciding L such that the number of moves used when M is executed on an input string ω is in $O(f(|\omega|))$, for all $\omega \in \Sigma^\star$.
- This can (trivially) be considered as a ***nondeterministic*** Turing machine $\widehat{M}$ (that never does any "guessing"), with language L, such that the computation tree for $\widehat{M}$ and a string ω has depth in $O(f(|\omega|))$, for all $\omega \in \Sigma^\star$. It follows that $L \in$ NTIME(f).
- Since L was arbitrarily chosen from TIME(f) it follows that TIME(f) $\subseteq$ NTIME(f), as claimed. $\square$

Relating Nondeterministic Computation to Deterministic Computation

Claim #3: For every function $f : \mathbb{N} \to \mathbb{N}$ and for every language $L \subseteq \Sigma^\star$ such that $f \in \mathsf{NTIME}(f)$, there exists an integer constant c (depending on L) such that $L \in \mathsf{TIME}(c^f)$. Thus

$$\mathsf{NTIME}(f) \subseteq \bigcup_{c \in \mathbb{N}} \mathsf{TIME}(c^f).$$

Idea of Proof: Let $L \subseteq \Sigma^\star$ such that $L \in \mathsf{NTIME}(f)$.

- Then there exists a nondeterministic Turing machine M deciding L such that the computation tree for M and ω has depth in $O(f(|\omega|))$ for every string $\omega \in \Sigma^\star$.

Relating Nondeterministic Computation to Deterministic Computation

- It follows that there exists an integer constants[1] $\widehat{c}$ and d such that the ***size*** of the computation tree for M and ω is at most $d \times \widehat{c}^{f(|\omega|)}$ for all $\omega \in \Sigma^\star$.
- It is possible to use a deterministic multi-tape Turing machine $\widehat{M}$ that decides L by performing a ***depth-first search*** on the computation tree for M and ω, using a number of steps in $O\left(c^{f(|\omega|)}\right)$, for all $\omega \in \Sigma^\star$, when $c = \widehat{c} + 1$ — establishing the claim. □

[1] These depend on M, and the precise bound on the depth of the computation tree that can be obtained.

Relating Nondeterministic Computation to Deterministic Computation

Definition:

$$\mathcal{NP} = \bigcup_{k \geq 1} \text{NTIME}(n^k).$$

It follows by Claims #2 and #3 that

$$\mathcal{P} \subseteq \mathcal{NP} \subseteq \textit{EXPTIME} = \bigcup_{k \geq 1} \text{TIME}(2^{(n^k)}).$$

This is (just about) all that has been ***proved*** about the relationship between deterministic time and nondeterministic polynomial time.

Relating Nondeterministic Computation to Deterministic Computation

Cook's Conjecture: $\mathcal{P} \neq \mathcal{NP}$.

- Future lectures will concern the use of reductions to establish additional (quite significant) results about the relationship between $\mathcal{P}$ and $\mathcal{NP}$, assuming that Cook's conjecture is correct.