CS221: Computational Complexity	Prof. Salil Vadhan
Lecture 6: Nondeterministic Time	
Oct. 2, 2002	*Scribe: Arthur Rudolph*

Contents

1 Nondeterministic Time

- $\mathbf{NTIME}(f(n))$ leads to $\mathbf{NP}, \mathbf{NEXP}$
- $\mathbf{NSPACE}(f(n))$ leads to $\mathbf{NL}, \mathbf{NPSPACE} \longleftarrow$ *next week*

For a complexity class $\mathcal{C}$:

$$\mathbf{co}\text{-}\mathcal{C} = \{\bar{L} : L \in \mathcal{C}\} (\neq \bar{\mathcal{C}})$$

It is clear that $\mathbf{P} = \mathbf{co}\text{-}\mathbf{P}, \mathbf{L} = \mathbf{co}\text{-}\mathbf{L}$, etc. (deterministic algorithms can flip their answer), but $\mathbf{NP} \stackrel{?}{=} \mathbf{co}\text{-}\mathbf{NP}, \mathbf{NL} \stackrel{?}{=} \mathbf{co}\text{-}\mathbf{NL}, \mathbf{NPSPACE} \stackrel{?}{=} \mathbf{co}\text{-}\mathbf{NPSPACE}$ are intriguing questions. Problems in $\mathbf{co}\text{-}\mathbf{NP}$:

$$\textsc{Unsatisfiability} = \{\varphi : \neg\exists a \text{ s.t. } \varphi(\bar{a}) = 1\}$$
$$\textsc{Tautology} = \{\varphi : \forall \bar{a} \varphi(\bar{a}) = 1\}$$

If $\mathbf{NP} = \mathbf{co}\text{-}\mathbf{NP}$, then every tautology has a short proof. (This observation has led to an entire area, called *Proof Complexity*, that studies the lengths of proofs.)

Proposition 1 *If* $\mathbf{NP} \neq \mathbf{co}\text{-}\mathbf{NP}$ *then* $\mathbf{P} \neq \mathbf{NP}$

1.1 What about NEXP?

$$\textsc{Exp-Proof} = \{(\varphi, \underbrace{n}_{\text{binary}}) : \varphi \text{ is math. thm. with a proof of length} \leq n\} \in \mathbf{NEXP}$$

$\mathbf{NEXP}$ is like $\mathbf{NP}$ but with succinct inputs (equivalently, with exponentially long certificates).

Proposition 2 $\mathbf{EXP} \neq \mathbf{NEXP} \Rightarrow \mathbf{P} \neq \mathbf{NP}$

We prove this using a simple but useful technique called *padding*.

Proof: We prove the contrapositive: $\mathbf{P} = \mathbf{NP} \Rightarrow \mathbf{EXP} = \mathbf{NEXP}$. Assume $\mathbf{P} = \mathbf{NP}$. Let $L \in \mathbf{NEXP}$. Then it is decided by a NTM M running in time 2^{n^c}. Define

$$L_{pad} = \{(x, 1^{2^{|x|^c}}) : x \in L\}$$

Claim 3 $L_{pad} \in \mathbf{NP}$

> **Proof of claim:** On input $(x, 1^k)$, check $k = 2^{|x|^c}$, if not reject. Otherwise, simulate M on x and behave the same. $\square$

By our assumption, we then have $L_{pad} \in \mathbf{P}$, with poly-time alg A.

Claim 4 $L \in \mathbf{EXP}$

> **Proof of claim:** Given x, run $A((x, 1^{2^{|x|^c}})) \leftarrow$time poly $(|x|, 2^{|x|^c}) \leq 2^{|x|^{c'}}$ $\square$

$\blacksquare$

More generally, with padding we can prove:

Proposition 5 *If $f(n), g(n) \geq n$ proper, then*

$$\mathbf{TIME}(f(n)) = \mathbf{NTIME}(f(n)) \Rightarrow \mathbf{TIME}(f(g(n))) = \mathbf{NTIME}(f(g(n)))$$

Padding is even more general:

- Applies to any pair of resources, bounds need not be same.
- "Equality (or inclusion) translates upwards, inequality (or separation) translates downwards."

2 Nondeterministic Time Hierarchy

Such a thing exists even though we can't flip output (don't know if closed under complement).

Theorem 6 *If $f(n), g(n) \geq n$ proper, $g(n) = \omega(f(n))$ then* $\mathbf{NTIME}(f(n)) \subsetneq \mathbf{NTIME}(g(n))$

Proof Sketch: For $f(n) = n^2, g(n) = n^3$. The technique we'll use is called "Lazy Diagonalization." Divide $\mathbb{N}$ into huge intervals, define $n_1 = 1$ and $n_{i+1} = 2^{n_i^4}$ for $i > 1$. Define $I_i = [n_i, n_{i+1}]$. Construct n^3 time NTM M, diagonalize against M_i somewhere in I_i.

$M(x)$:

- If $x \notin 1^*$, reject.
- If $x = 1^n$, say $n \in I_i$:
  - If $n < n_{i+1}$, simulate Mi on 1^{n+1} for $(n+1)^2$ steps and do the same.
  - If $n = n_{i+1}$, compute whether M_i accepts or rejects 1^{n_i} (within n_i^2 steps) and do opposite. Do this by enumerating over all $2^{\mathcal{O}(n_i^2)} \leq \mathcal{O}(n_{i+1})$ computations of length n_i^2.

Suppose $\exists$ time n^2 NTM M_i s.t. $L(M_i) = L(M)$.

$$M_i(1^{n_i}) = M(1^{n_i}) = M_i(1^{n_{i+1}}) = M(1^{n_{i+1}}) = M_i(1^{n_{i+2}}) = \dots$$

but also by the construction we have $M(1^{n_{i+1}}) \neq M(1^{n_i})$ for some i. $\Rightarrow\Leftarrow$. $\square$

Remark 7 Without properness conditions weird things happen, e.g.

$$\exists f \text{ s.t. } \mathbf{TIME}(f(n)) = \mathbf{TIME}(2^{f(n)})$$

(Gap theorem (Papadimitriou, Thm. 7.3)).

3 Relationships between Resources

3.1 Time vs. Space

1. $\mathbf{TIME}(f(n)) \subseteq \mathbf{SPACE}(f(n))$. (Can't write $> f$ symbols in f steps). Thus $\mathbf{P} \subseteq \mathbf{PSPACE}$ [It known that $\mathbf{TIME}(f(n)) \subseteq \mathbf{SPACE}(\frac{f(n)}{\log f(n)})$, but it takes a lot of work to prove (Hopcroft-Paul-Valiant).]

2. $\mathbf{SPACE}(f(n)) \subseteq \cup_c \mathbf{TIME}(2^{cf(n)})$, provided $f(n) \geq \log n$. This is because we can bound the number of "configurations" of a space $f(n)$ algorithm as follows:

$$(\#\text{ of states}) \cdot \underbrace{\mathcal{O}(|\Sigma|)^{f(n)}}_{\text{tapes,cursors,etc.}} \cdot \underbrace{n}_{\text{position}} \leq 2^{\mathcal{O}(f(n))}$$

   (Because of that n term we need $f(n) \geq \log n$.) If the algorithm takes more than this many steps, a configuration must have been repeated and hence the algorithm is in an infinite loop and will never halt. We can thus conclude that $\mathbf{L} \subseteq \mathbf{P}$ and $\mathbf{PSPACE} \subseteq \mathbf{EXP}$.