Hierarchy theorems

- We now see that given strictly more resources (eg. time, space, nondeterminism) TMs can solve strictly <u>more</u> problems.

- A common feature in the proofs is <u>diagonalization</u>.

<u>Theorem 1</u>: If $g(n) = \omega(f(n) \cdot \log f(n))$ then $\mathrm{Dtime}(f(n)) \subsetneq \mathrm{Dtime}(g(n))$.

<u>Proof</u>:

- Let us design a TM, in the RHS, that is different from each one in the LHS.

- Consider the TM D : On input x,
  (1) If x is not a TM description then <u>output</u> 0.
  (M_x is the TM described by x)
  (2) Else simulate $M_x(x)$ for $g(|x|)$ steps:

(2.1) If it doesn't halt then <u>output</u> 0.
(2.2) Else <u>output</u> $1 - M_x(x)$.

- By definition, D decides a language $L \in \mathrm{Dtime}(g(n))$.

- Is $L \in \mathrm{Dtime}(f(n))$? Suppose yes. Let M be a TM deciding L in time $c \cdot f(n)$, for all $n \geq n_0$.
 (c & n_0 are some constants)

- Pick a "large" string y describing M s.t. $g(|y|) > d \cdot f(|y|) \cdot \log f(|y|)$, for $|y| \geq n_0$.
 (where d is the constant s.t. the "universal" TM simulates $M_y(y)$ in time $d \cdot f(|y|) \cdot \log f(|y|)$.)

- What is $D(y)$?
- Note that $M_y(y) = M(y)$ runs for time $c \cdot f(|y|)$ & halts.

- Thus, D halts on y, in time $d \cdot f(|y|) \cdot \log f(|y|) < g(|y|)$, and <u>outputs</u> $1 - M(y)$.

- This contradicts that M decides L!
$\Rightarrow$ M does not exist.
$\Rightarrow$ $\mathrm{Dtime}(f(n)) \subsetneq \mathrm{Dtime}(g(n))$. $\square$

<u>Space hierarchy</u>

<u>Defn</u>: <u>Space</u> $(f(n)) := \{L \mid L$ is decided by a TM that use $O(f(n))$ space $\}$.

<u>Theorem 2</u>: If $g(n) = \omega(f(n))$ then
$$\mathrm{Space}(f(n)) \subsetneq \mathrm{Space}(g(n)).$$

<u>Proof</u>:

- Again, we define a TM D as before.
- Further, note that the universal TM can simulate $M_y(y)$ in roughly the <u>same</u> space

as is the space-complexity of the TM y. $\square$

Open: A result as strong as Thm 2 for the time hierarchy?

ND Time hierarchy

- The proof of nondeterministic time hierarchy is quite involved.
- The issue is negation: For an NDTM M, we do not know whether the computation $1 - M(x)$ can be done by a "fast" NDTM.

Theorem 3: If $g(n) = \omega(f(n))$ then
$$\mathrm{Ntime}(f(n)) \subsetneq \mathrm{Ntime}(g(n)).$$

<u>Proof:</u> • The idea is to design a TM D, in the RHS, that differs with the LHS <u>very</u> rarely. (so, negation requires few nondet. bits.)
This is called <u>lazy diagonalization</u>.

• For this purpose we need a very rapidly growing function $s : \mathbb{N} \to \mathbb{N}$ s.t. $g(s(i+1)) \geq s(i+1) \geq 2^{g(s(i)+1)}$, for all $i \in \mathbb{N}$.

• Consider the NDTM D: On input x,

(1) If $x \notin 1^*$, then <u>output</u> 0.

M_i is the NDTM described by i

(2) If ($x = 1^n$ & $s(i) < n < s(i+1)$) then simulate $M_i(1^{n+1})$ for $g(n)$ steps.

(3) If ($x = 1^n$ & $n = s(i+1)$) then <u>output</u> 1 iff $M_i(1^{1+s(i)})$ rejects in $g(1+s(i))$ steps.

• Clearly, D is an NDTM with time

Complexity (for $n = s(i+1)$) being:

$$2^{g(s(i)+1)} \leq s(i+1) \leq g(s(i+1)) = g(n).$$

$\Rightarrow$ D decides a language $L \in \text{Ntime}(g(n))$.

- Say, an NDTM M decides L in time $f(n) = o(g(n))$.

  Pick a "large" j s.t. $M = M_j$.

- By the definition of D (step-(2)):

  $\forall n \in (s(j), s(j+1)),\ D(1^n) = M_j(1^{n+1})$.

$\Rightarrow$

$\forall n \in (s(j), s(j+1)),\ M_j(1^n) = M_j(1^{n+1})$.

$\Rightarrow$

$M_j(1^{s(j)+1}) = M_j(1^{s(j+1)}) = D(1^{s(j+1)})$.

- But by step-(3) of D:

  $D(1^{s(j+1)}) \neq M_j(1^{s(j)+1})$.