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>. [Hartmanis, Stearns, Lewis 1965]

<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 output 0.
(2.2) Else output $1 - M_x(x)$.

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

- Is $L \in Dtime(f(n))$? 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 [Cook '73] [Zak '83]

- 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 very rarely. (so, negation requires few nondet. bits.)
This is called lazy diagonalization.

• 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 $i \in \mathbb{N}$.

$s(i)$ is like a tower of 2's.

• Consider the NDTM D: On input x,

(1) If $x \notin 1^*$, then output 0.

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

M_i is the NDTM described by i

(3) If ($x = 1^n$ & $n = s(i+1)$) then output 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:

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

implement (3) as a TM

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

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

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

  ($\Rightarrow c \cdot f(n+1) < g(n)$, for $n > s(j)$.)

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

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

$\because L(M) = L(D) = L(M_j)$

$\Rightarrow$

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

$\Rightarrow$

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

- But by step-(3) of D:

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

- This contradiction refutes the existence of M.
  $\Rightarrow \mathrm{Ntime}(f) \subsetneq \mathrm{Ntime}(g)$. $\square$

- We continue with more diagonalization proofs.

- Are <u>all</u> the problems in $NP \setminus P$, NP-complete?

<u>Ladner's theorem</u>[1975]<u>:</u> If $P \neq NP$ then $\exists L \in NP \setminus P$ that is not NP-complete.

<u>Proof</u>:

- <u>Idea</u>: Pad SAT & use diagonalization.

- Say, $P \neq NP$. Then $SAT \notin P$. For some fn. $H(\cdot)$ consider the padding:
  $SAT_H := \{\phi 0 1^{n^{H(n)}} \mid \phi \in SAT \ \& \ |\phi| = n\}$.

▷ $H(n) \to \infty \Rightarrow SAT_H$ is not NP-Complete.

<u>Pf</u>:-

If $SAT \leq_p SAT_H$ & $H(n) \to \infty$, then a CNF ψ of size n reduces to an instance $\phi 0 1^{|\phi|^{H(|\phi|)}}$ of size n^c (constant c).

$\Rightarrow |\phi| + |\phi|^{H(|\phi|)} = O(n^c)$.

$\Rightarrow |\phi| = o(n)$.

Thus, ψ of size n reduces to a ϕ of size $o(n)$.

On repeating this again & again, we get a CNF τ of size $O(1)$.

$\Rightarrow SAT \in P$, which is a contradiction.

□

• To deduce $SAT_H \notin P$ we define H in a way so that it grows <u>very</u> <u>slowly</u>:

$H(n)$ is the smallest $i < \lg\lg n$ s.t. $\forall x \in \{0,1\}^{\leq \lg n}$, M_i accepts x in time $\leq i \cdot |x|^i$ iff $x \in SAT_H$, ← recursive defn.

Or, if there is no such i then $H(n) := \lg\lg n$.

• How easy is it to compute $H(n)$?

By "brute-force" it requires

$$\underset{\uparrow}{\lg\lg n} \times \underset{\uparrow}{2^{\lg n}} \times \underset{\uparrow}{(\lg n)^{\lg\lg n}} \times \underset{\uparrow}{2^{\lg n}} = o(n^3).$$

i's   # x's   # M_i steps   solving SAT on $\lg n$ size

▷ $SAT_H \in NP$.

▷ $SAT_H \notin P$.

Pf: Suppose a TM M solves SAT_H in time $\leq c \cdot n^c$. Pick a $j > c$ s.t. $M = M_j$.

$\Rightarrow$ M_j decides SAT_H in $< n^j$ time, implying $H(n) \leq j$, $\forall n > 2^{2^j}$.

$\Rightarrow$ SAT_H is just SAT padded with n^j 1's.

$\Rightarrow$ $SAT \in P$. A contradiction. □

▷ $H(n) \to \infty$.

Pf: Since $SAT_H \notin P$, $\forall i \exists x$ s.t. M_i cannot decide $x \in ? SAT_H$ in time $i \cdot |x|^i$.

$\Rightarrow$ $H(n) \neq i$, $\forall n > 2^{|x|}$.

$\Rightarrow$ $H(n)$ takes a value i only for finitely many n. $\square$

- Thus, we have a poly-time fn. H s.t. $SAT_H \in NP \setminus P$ & SAT_H is not NP-c. $\square$

– We have seen such clever diagonalization tricks. Could they show $P \neq NP$?