Hierarchy Theorems

11 March, 2021 13:13

$\mathrm{DTIME}(t(n)) = \{$ languages / problems solvable in time $O(t(n))\}$

$\mathrm{NTIME}(t(n)) = \{$ –"– solvable nondeterministically in time $O(t(n))\}$

$P = \bigcup_k \mathrm{DTIME}(n^k)$ $\quad NP = \bigcup_k \mathrm{NTIME}(n^k)$

$E = \bigcup_k \mathrm{DTIME}(2^{kn})$ $\quad NE = \bigcup_k \mathrm{NTIME}(2^{kn})$

$EXP = \bigcup_k \mathrm{DTIME}(2^{n^k})$ $\quad NEXP = \bigcup_k \mathrm{NTIME}(2^{n^k})$

$EEXP = \bigcup_k \mathrm{DTIME}(2^{2^{n^k}})$ $\quad NEEXP = \bigcup_k \mathrm{NTIME}(2^{2^{n^k}})$

<u>We will see</u>: $P \subsetneq E \subsetneq EXP \subsetneq EEXP \ldots$

"more time" = "more solvable problems"

- $T(n)$ is time <u>constructible</u> if there exists a TM M which on input 0^n halts exactly after $T(n)$ steps.

<u>Thm</u> (time hierarchy): Let $t(n), T(n) \geq n$ be time constructible & $T(n) \in \omega(t(n) \log t(n))$. Then $\mathrm{DTIME}(t(n)) \subsetneq \mathrm{DTIME}(T(n))$.

<u>Pf</u>: 1) $\mathrm{DTIME}(t(n)) \subseteq \mathrm{DTIME}(T(n))$.

2) to show $\mathrm{DTIME}(t(n)) \neq \mathrm{DTIME}(T(n))$

Pf: 1) $DTIME(t(n)) \subseteq DTIME(T(n))$.

2) to show $DTIME(t(n)) \neq DTIME(T(n))$

we will construct a language $L \in DTIME(T(n))$

s.t. $L \notin DTIME(t(n))$.

L is described by its algorithm.

The algorithm diagonalizes against all machines

working in time $O(t(n))$.

Alg. For L:

on input x of length n

let n be in binary : $1{*}{*}{*}{*}\cdots{*}1\overbrace{00\cdots0}^{i}$.

simulate TM M_i on x for $T(n)$

steps of the simulation using a universal TM.

If M_i accepts the input → REJECT x

o/w ACCEPT x.

end.

Clearly $L \in DTIME(T(n))$

$L \notin DTIME(t(n))$.:

let M_i be any machine running in time $O(t(n))$.

For n large enough the time taken by

the universal TM to simulate M_i on input

of length n is $\leq T(n)$. (Since $t(n) \cdot \log(t(n)) \in o(T(n))$)

on input x of large enough length

of length ''

On any input x of large enough length $n = 10\ldots01\overbrace{000\ldots0}^{i}$, the simulation of M_i on x in the above alg. finishes & so

$x \in L(M_i)$ iff $x \notin L$.

Hence $L(M_i) \neq L$.

$\Rightarrow L \notin DTIME(t(n))$. ∎

<u>Thm</u> (nondeterministic time hierarchy)

Let $t(n), T(n) \geq n$ be time constructible fcn's s.t. $T(n) \in \omega(t(n+1))$. Then $NTIME(t(n)) \subsetneq NTIME(T(n))$.

<u>Ex</u>:
- $n \lg\lg\lg n \in \omega(n+1)$
- for any polynomial $p(n)$, $p(n+1) \in O(p(n))$
- but $2^{n^2} \in o(2^{(n+1)^2})$

$$2^{(n+1)^2} = 2^{n^2+2n+1} = 2^{n^2} \cdot 2^{2n+1}$$

<u>Pf</u>: <u>Delayed diagonalization</u>

Define $n_0, n_1, n_2, \ldots$ inductively

$n_0 = 1$ & $n_k = 2^{T(n_{k-1}+1)}$

For inputs of length $n \in (n_{k-1}, n_k]$ we will diagonalize the same machine M_i

we will diagonalize the same machine M_i
running in time $t(n)$.
Construction of $L \in DTIME(T(n)) \setminus DTIME(t(n))$

Alg. for L:

on input 0^n
find k s.t. $n_{k-1} < n \leq n_k$
let $k = 1***\ldots 1\underbrace{00\ldots0}_{i}$

If $n < n_k$ then simulate NTM M_i on
input $0^{(n+1)}$ using nondet. universal TM
running for $T(n)$ steps.
if the simulation accepts $\to$ ACCEPT 0^n,
o/w REJECT 0^n.

If $n = n_k$ deterministically check in time $T(n_k)$,
whether M_i accepts $0^{n_{k-1}+1}$ in $T(n_{k-1}+1)$ steps.
If M_i accepts $0^{n_{k-1}+1}$ then ACCEPT 0^n
o/w REJECT 0^n.

this check takes time

$$\leq 2^{T(n_{k-1}+1)} \cdot \left(T(n_{k-1}+1)\right)^2$$

$$\leq T(n_k) \quad \text{for } n_k \text{ large enough.}$$

$\leq T(n_k)$ for n_k large enough.

- If M_i runs in time $O(t(n))$ then for all large enough n its running time is bounded by $T(n)$.

$L \in DTIME(T(n))$. We claim $L \notin DTIME(t(n))$.

Let M_i runs in time $O(t(n))$.

<u>3 cases:</u>

1) M_i accepts 0^n for all $n_{k-1} < n \leq n_k$ where $k = 1{*}{*}\cdots 1\underbrace{000\cdots 0}_{i}$ is large enough.

   then $0^{n_k} \notin L$. $\Rightarrow L \neq L(M_i)$

2) M_i rejects 0^n for all $n_{k-1} < n \leq n_k$

   then $0^{n_k} \in L$. $\Rightarrow L \neq L(M_i)$

3) M_i accepts 0^n & rejects 0^{n+1} for some $n_{k-1} < n \leq n_k$.

   then $0^{n+1} \in L \Rightarrow L \neq L(M_i)$

   or M_i rejects 0^n & accepts 0^{n+1} for some $n_{k-1} < n \leq n_k$

   then $0^{n+1} \notin L \Rightarrow L \neq L(M_i)$

$\Rightarrow NP \subsetneq NE \subsetneq NEXP \subsetneq NEEXP$

$\Rightarrow NP \underset{\neq}{\subseteq} NE \underset{\neq}{\subseteq} NEXP \neq NEEXP$

Claim: If $NP = NEXP \Rightarrow NEXP = NEEXP$
$\Rightarrow NP = NEEXP$

Corollary: $NP \underset{\neq}{\subseteq} NEXP$ since $NP = NEXP$ &
$NP \subseteq EXP \underset{\neq}{\subseteq} EEXP$
$\subseteq NEEXP$
are in contradiction.

Pf: $\forall L \in NEEXP$ want to show $L \in NEXP$
assuming $NP = NEXP$.

"padding argument"

Let L be accepted by NTM M running in time $2^{2^{n^k}}$.

Def. $L' = \{ x \# 0^{2^{|x|^k}} ; x \in L \}$

$L' \in NEXP$: on input $y = x \# 0^{\ell}$ check that the number of padded 0's is correct & if it is, run M on x.

$\rightarrow$ takes time $\leq O(2^{|y|}) = O(2^{2^{|x|^k}})$

by our assumption $NEXP = NP$ so
$L' \in NP$ ... recognized by NTM M' in time $n^{k'}$.

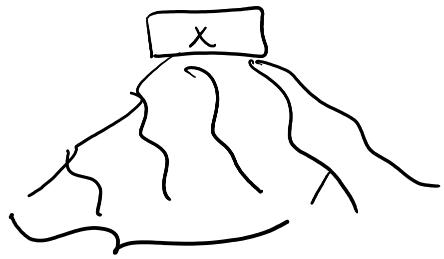
$\Rightarrow L \in NEXP$: on input x, create $x \# 2^{|x|^k}$ & run M' on it.

$\rightarrow$ this takes time $\left(2^{|x|^k}\right)^{k'}$.

$\boxtimes$

BPTIME$(t(n))$ problems solvable by randomized

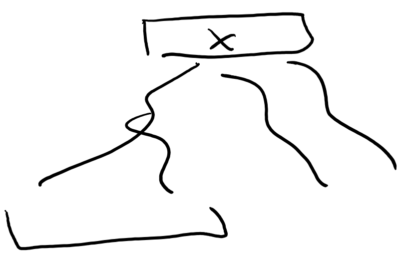
BPTIME$(t(n))$ … problems solvable by randomized algorithms running in time $O(t(n))$ with error $\leq \frac{1}{3}$.



computation tree

$\geq 2/3$ correct outputs

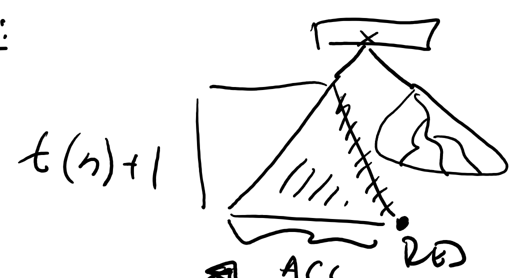
PTIME$(t(n))$ … problems solvable by randomized alg. running in time $O(t(n))$ with error $< \frac{1}{2}$.



$> \frac{1}{2}$ correct outputs

- $NP \subseteq PP$ (Toda's thm: $PH \subseteq P^{PP}$)

Pf:



nondet comp. in time $t(n)$

prob. $\frac{1}{2} - 2^{-t(n)-2}$ of ACCEPT

it suffices a single accepting branch

of the nondet. computation to give over 1/2 prob. $\square$

- PRIMES $\in$ BPP

POLYNOMIAL IDENTITY TESTING $\in$ BPP

$$BPP = \bigcup_k BPTIME(n^k).$$

Schwarz-Zippel lemma: Let $p \not\equiv 0$ be a polynomial in n variables of total degree $\leq d$ over a field $\mathbb{F}$. Let $S \subseteq \mathbb{F}$. The # of roots of p in S^n is at most $d \cdot |S|^{n-1}$.

Pf: Induction on n

1) $n=1$ ... standard fact about polynomials
2) Induction step $n-1 \to n$

$$p(x_1 \ldots x_n) = \sum_{i=0}^{d} p_i(x_1 \ldots x_{n-1}) \cdot x_n^i$$

Let d' be the largest i s.t. $p_i \not\equiv 0$.

$\deg p' \leq d-d'$

For $a_1, \ldots, a_n \in S^n$ be a root of $p(\ldots)$

either $p_{d'}(a_1 \ldots a_{n-1})$ is zero (which happens for at most $(d-d')|S|^{n-2}$ by I.H. $(n-1)$-tuples $a_1, \ldots a_{n-1}$) or

a_n is a root of $p_{a_1 \ldots a_{n-1}}(x_n) = \sum_{i=0}^{d'} p(a_1 \ldots a_{n-1}) x_n^i$

(which happens for at most d' values of a_n)

Hence in total, there are at most

$$\leq (d-d')\cdot |S|^{n-2}\cdot |S| + d'\cdot |S|^{n-1} = d\cdot |S|^{n-1}.$$

$\boxed{B}$

Computation with advice

$M \dots$ TM $\qquad g: \mathbb{N} \to \{0,1\}^*$ advice fcn

$x, g(|x|) \dots$ input to machine M

different advice fcn's $\to$ different languages computed by M

$BPTIME(t(n))/f(n) \dots$ problems solvable by randomized TM's in time $O(t(n))$ with error $\leq \frac{1}{3}$ & advice fcn g s.t. $|g(n)| \leq f(n)$.

the restriction on the error is only required to hold on correct advice. On incorrect advice the error might be arbitrary.

$BPTIME(n)/1 \dots$ contains non-recursive languages, e.g. unary encoding of the Halting pblm.

- For a proof of time hierarchy thm using diagonalization need to be able to enumerate all machines

we need
with error bounded by 1/3. We do not know,
how to do that.

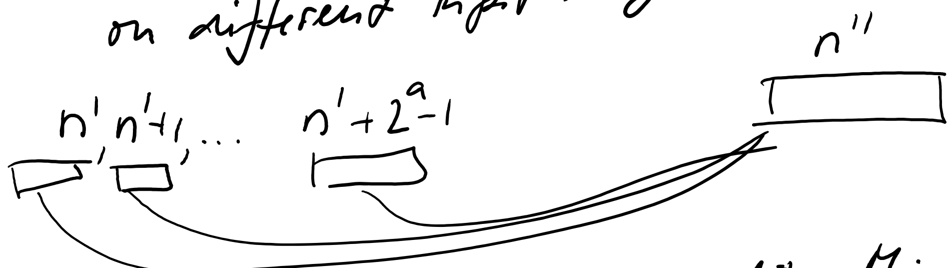
Open Question: Is there a complete problem for BPP?

Thm: $\forall$ const. $a \geq 2, k \geq 1$, $\exists L \in BPTIME(n^{4^{ak}})/1$

s.t. $L \notin BPTIME(n^k)/a$

Pf: Delayed diagonalization

- diagonalization against all machines with all possible advice.
- the advice bit tells us, whether it is safe to simulate machine M_i with advice z on some inputs.
- different advice strings to M_i will be diagonalized on different input lengths $\qquad n'' \approx (n')^2$



- on inputs of length $n'+j$ simulate M_i on some inputs of length n'' with advice j. The advice bit for our machine tells us whether this is safe.
- since $n'' \approx (n')^2$ after $\log n$ iterations we have long enough input to perform.

delayed diagonalization.

def: $n_0^* = 1$

$n_i = n_{i-1}^* + 1$

$m_i = \lg n_i$

$n_i^* = n_i^{d^{m_i}}$

$d = 2^{(a+1)k}$

notice: $n_i^* \geq \left(2^{(n_i^{a+1})^k}\right)^2$

so on input of length n_i^* there is enough time to determine behaviour of M_i on inputs of length $\leq n_i + (2^a)^{m_i} = n_i^a + n_i$

def: $n_{i,j} = n_i^{d^j}$ $\quad j = 0, \dots, \lg n_i$

$\forall\, w \in \{0,1\}^{a(m_i - j - 1)}$ & $z \in \{0,1\}^a$, $j < \lg n_i$

$n_{i,j,wz} = n_{i,j} + \overline{wz}$

↳ # represented by wz in binary

for strings y of length $n_{i,j,wz}$ define

$f(y) = yz\,1\,0^{n_{i,j+1,w} - n_{i,j,wz} - a - 1}$

$\Rightarrow |F(y)| = n_{i,j+1,w}$

alg for M: on input x of length n & advice $b \in \{0,1\}$

1. If $b = 0$ $\longrightarrow$ REJECT
2. If $n = n_{i,j,wz}$ for some wz of appropriate size, run M_i on input $f(x)$ & advice z.

n.

3. If $n = n_i^*$, check behavior of M_i on input y s.t. $|y| \le n_i + 2^{am_i} \le n_i^{a+1}$

$$\underbrace{f(f(f(\dots f(y))\dots)}_{m_i\text{-times}} = x$$

& use the first a bits of y as an advice for M_i on y. ACCEPT $\Leftrightarrow$ M_i rejects y with the advice.

- It is easy to check that M has enough time to check the computation of M_i on input y running in time $|y|^k$.

running time of M:

step 2: $|f(x)| = n_{i,j+1,w} = n_i^{d^{j+1}} + \overline{w} = \left(n_i^{d^j}\right)^d + \overline{w}$

$$\le \left(n_i^{d^j} + \overline{w}\right)^d \le \left(n_{i,j,w}\right)^d$$

running M_i on $f(x)$ requires time

$O\left(|f(x)|^k \cdot \lg |f(x)|^k\right)$ ... needs to simulate M_i using universal TM

constant depends on M_i

$\wedge$

$(n_{i,j,w})^{2dk}$ For n large enough constant depends on M_i $\quad (n_{i,j,wz})^{2dk} \ldots$ For n large enough relative to M_i ($\Rightarrow$ need M_i to be diagonalized infinitely often)

$$\text{since } 2dk = 2 \cdot 2^{(a+1)k} \cdot k = 2^{ak} \cdot 2 \cdot 2^{k} \cdot k \underset{a \geq 2}{\leq} 4^{ak}$$

$$(n_{i,j,wz})^{2dk} \leq (|x|)^{4^{ak}} \checkmark$$

<u>Step 3</u>: See notice above - $n_i^* \geq \left(2^{(n_i^{a+1})^k}\right)^2$ ✓

$\Rightarrow$ M runs in time $O(n^{4^{ak}})$

<u>M differs from each M_i that runs in time $O(n^k)$:</u>

Let M_i run in time $O(n^k)$ with advice sequence

$$z_1, z_2, \ldots, z_n, \ldots$$

want to show that language M differs from M_i with such advice sequence.

- pick $x = \underbrace{* * * \quad \cdots \quad *}$ of length n_i^*

so that $y_0, y_1, y_2, \ldots y_{m_i}$ ($y_{m_i} = x$) defined inductively $j = m_i, \ldots, 1$

$f(y_{j-1}) = y_j$ satisfy:

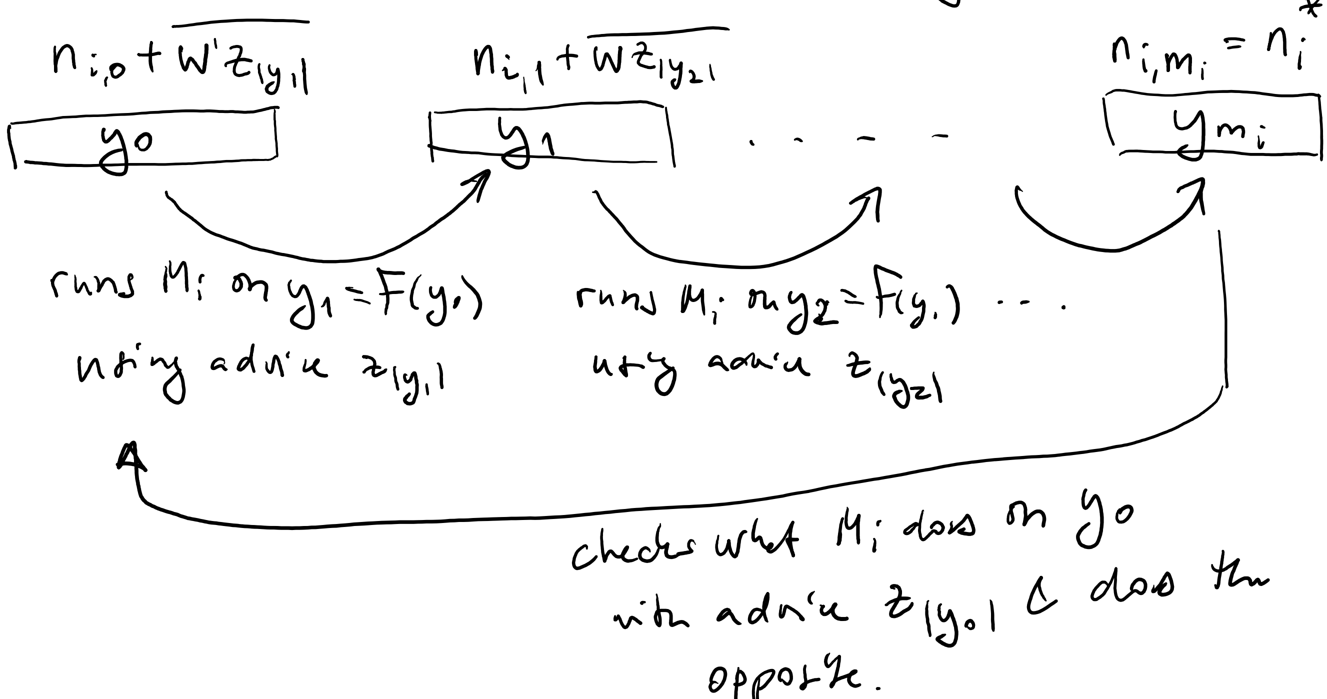
if $|y_{j-1}| = n_{i,j-1} + \overline{wz}$ $\qquad (*)$

$$\text{if } |y_{j-1}| = n_{i,j-1} + \overline{wz} \qquad (*)$$

$$\text{then } z = z_{|y_j|}$$

$\to$ one has to determine x from right so that $(*)$ would be satisfied.

Furthermore let $y_0 = z_{|y_0|} 00 \cdots 0$



1) If M_i accepts all the $y_0, \dots y_{m_i}$ then

M on y_{m_i} differs from M_i on y_{m_i}.

2) if M_i rejects all the $y_0 \dots y_{m_i}$ then

— " —

3) Otherwise M_i accepts y_i & rejects y_{i+1}. But then

M rejects y_i (or vice versa M_i rej. y_i & acc y_{i+1})

$\to$ M differs from M_i that uses advice $z_1, z_2, \dots$ $\boxed{Z}$