CSE431 Theory of Computing
Lecturer : James Lee
Scriber : Bryan Lu
Lecture 17, May 27, 2014

Hierarchy Theorems

Overview:

$L \subseteq P \subseteq NP \subseteq PSAPCE \subseteq PSPACE = NPSPACE \subseteq EXPTIME$

- L : Logarithmic Space
- Now, different resources are not comparable.
- By Hierarchy Theorems,
  1. $P \neq EXPTIME$
  2. $L \neq PSPACE$

Definition:

Space Constructible Function:

$f : \mathbb{N} \rightarrow \mathbb{N}$ is space constructible
if $f(n) \geqslant c(\log(n))$ and there is a TM that computes the map

111...1 -> f(n) is binary in O(f(n)) space
(n 1s)

Time Constructible Function:

$t : \mathbb{N} \rightarrow \mathbb{N}$ is time constructible
if $t(n) \geqslant c(\log(n))$ and there is a TM that computes the map

11111.....1 ->f t(n) is binary Using O(f(n)) time
(n 1s)

SPACE Hierarchy Theorem:

Definition:

For any space constructible function $f : \mathbb{N} \rightarrow \mathbb{N}$, there is a language A such that A can be decided in O(f(n)) space, but not in o(f(n)) space.

Corollary:

$SPACE(n^k) \not\subseteq SPACE(n^{(k+1)})$ for all k

Informal: (Using Diagonalization)

"D takes <M> as input
n = length(<M>)
Simulate M on <M> in f(n) space
Count the # of steps up to $2^{(f(n))}$
if M(<M>) tries to use > f(n) space, then REJECT
if M accepts, we REJECT
if M rejects, we ACCEPT"

Claim :

- D runs in O(f(n)) space.
- D always halts
- A = L(D) can be decided in O(f(n)) space
- Suppose M uses o(f(n)) space, then

$$\text{D(<M>)} \neq \text{M(<M>)} \Rightarrow M \textit{ does not decides } A$$

Formally:

D = "On input w:

1. n = length(w)

1.5. Mark off f(n) space on the tape (Using the fact that f is space constructible)

2. if $w \neq \text{<M>}0\,1^*$ for some TM M, REJECT
3. Simulate M on w for $2^{(f(n))}$ steps
4. If M on w tries to use more then f(n) space or $2^{(f(n))}$ steps, REJECT
5. If M accepted w, REJECT
6. If M rejected w, ACCEPT"

Space Complexity of D:

1. Counting the length of w : O(log(n)) space = O(f(n)) space
2. Space for checking the equality $w \neq \text{<M>}0\,1^*$ requires O(log(n)) space
3. counter of 2^(f(n)) requires O(f(n)) space
4. Using some over head to simulate and check if it use over f(n) space for the simulation = O(f(n)) space

Proof :

(foot note : log(d)*g(n) bits)

Suppose M uses g(n) space and g(n) = o(f(n))

For some n_0 when $n \geqslant n_0$, we have g(n) < f(n)

if $|\text{<M>}| \geqslant n_0$ then $\text{D(<M>0)} \neq \text{M(<M>0)}$

if $|\text{<M>}| < n_0$ then $\text{D(<M>0}\,1^{(n_0)}) \neq \text{M(<M>0}\,1^{(n_0)})$

[Since $\text{<M>}0\,1^{(n_0)}$ is $n \geqslant n_0$ which g(n) < f(n)]

Hence $M \neq D$

Q.E.D.

TIME Hierarchy Theorem:

Definition:

For any time constructible function t(n),
there is a language A that can be decided in O(t(n)) time,
but not in o(t(n) / log(t(n))) time

Corollary:

TIME(n^k) <!= TIME(n^(k+1)) for all K >= 1

Informal:

D = "On input <M>,
Simulate M on <M> for [t(n) / log(t(n))] steps
Do the opposite of what M does"

Claims :

D runs in O(t(n)) time