Hierarchy Theorems

Peter Mayr

Computability Theory, April 14, 2021

So far we proved the following inclusions:

$$\mathsf{L} \subseteq \mathsf{NL} \subseteq \mathsf{P} \subseteq \mathsf{NP} \subseteq \mathsf{PSPACE} \subseteq \mathsf{EXPTIME} \subseteq \mathsf{EXPSPACE}$$

Question

Which are proper?

Space constructible functions

In simulations we often want to fix $s(n)$ space on the working tape before starting the actual computation without accruing any overhead in space complexity.

Definition

$s(n) \geq \log(n)$ is **space constructible** if there exists $N \in \mathbb{N}$ and a DTM with input tape that on any input of length $n \geq N$ uses and marks off $s(n)$ cells on its working tape and halts.

Note

Equivalently, $s(n)$ is space constructible iff there exists a DTM with input and output that on input 1^n computes $s(n)$ in space $O(s(n))$.

Example

$\log n, n^k, 2^n$ are space constructible.

E.g. $\log_2 n$ space is constructed by counting the length n of the input in binary.

Little o-notation

Definition

For $f, g \colon \mathbb{N} \to \mathbb{R}^+$ we say $f = o(g)$ (read f is little-o of g) if

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0.$$

Intuitively: g grows much faster than f.

Note

- $f = o(g) \Rightarrow f = O(g)$, but not conversely.
- $f \neq o(f)$

Separating space complexity classes

Space Hierarchy Theorem

Let $s(n)$ be space constructible and $r(n) = o(s(n))$.
Then $\text{DSPACE}(r(n)) \subsetneq \text{DSPACE}(s(n))$.

Proof.

Construct $L \in \text{DSPACE}(s(n)) \setminus \text{DSPACE}(r(n))$ by diagonalization.

Define DTM D such that on input x of length n:

1. D marks $s(n)$ space on working tape. Reject if D ever uses more space later.
2. If $x \neq \underbrace{0 \ldots 0}_{\text{padding}} \sharp(\text{M})$ for some DTM M, reject.
3. D simulates M on x within $s(n)$ space.
4. If M accepts x, D rejects (and conversely).

Space complexity of D:

- By the condition in step 1, D uses only $s(n)$ space but may loop on some x [Then $x \notin L(\mathrm{M})$ and $x \notin L(\mathrm{D})$].
- By adding a time counter to D we obtain a DTM D' that uses $O(s(n))$ space and halts on all inputs in $2^{O(s(n))}$ time such that $L(\mathrm{D}) = L(\mathrm{D'})$ (HW).

Thus $L(\mathrm{D})$ is computed in $O(s(n))$ space.

Claim: $L(\mathrm{D})$ is not computable in $o(s(n))$ space.

- Let M run in space $r(n)$ with $r(n) = o(s(n))$ and halt on all inputs.
- In step 3, depending on M's tape alphabet, $c \geq 1$ cells of D may be needed to encode a single cell of M. Hence D needs $cr(n)$ space to simulate M.
- Since $r(n) = o(s(n))$, $\exists N \; \forall n \geq N : \; cr(n) < s(n)$ and D's simulation runs to completion if M's input has length $\geq N$.
- Then $x := 0^N \sharp(\mathrm{M})$ is accepted by M iff x is rejected by D.
- Hence $L(\mathrm{D}) \neq L(\mathrm{M})$.

Consequences of the Space Hierarchy Theorem

Corollary

$\text{DSPACE}(n^k) \neq \text{DSPACE}(n^\ell)$ if $0 \leq k < \ell$.

Corollary

$\text{NL} \neq \text{PSPACE}$

Corollary

$\text{PSPACE} \neq \text{EXPSPACE}$

Proof.

HW $\square$

Time constructible functions

In simulations we often want to determine whether a TM has run for $t(n)$ steps without accruing any overhead in time complexity (i.e. in $O(t(n))$ time).

Definition

$t(n) \geq n \log(n)$ is **time constructible** if there exists a DTM with input tape that on input 1^n computes $t(n)$ in $O(t(n))$ time.

Example

$n \log n, n^k, 2^n \ldots$ are time constructible.

DTM to count the length of the input in binary:

- The counter increments by 1 (requiring $O(\log n)$ steps) for every input position.
- Running time is $O(n \log n)$.

It follows that $n \log n$ can be computed in binary in $O(n \log n)$ time.

Separating time complexity classes

Time Hierarchy Theorem

Let $t(n)$ be time constructible and $r(n) = o(\frac{t(n)}{\log t(n)})$.
Then $\mathsf{DTIME}(r(n)) \subsetneq \mathsf{DTIME}(t(n))$.

Proof.

Similar to Space Hierarchy Theorem but counting the number of steps in the simulation of M yields log-factor overhead.

Define a single tape DTM D such that on input x of length n:

1. Compute $t(n)$ and store $\frac{t(n)}{\log(t(n))}$ in a binary counter.
2. If $x \neq \underbrace{0 \ldots 0}_{\text{padding}} \sharp(\mathrm{M})$ for some DTM M, reject.
3. D simulates M on x. Before each step of M decrease time counter by 1; reject if 0 is reached.
4. If M accepts x, D rejects (and conversely).

Running time of D:

- Steps 1, 2 take $O(t(n))$ time since $t(n)$ is time constructible.
- To simulate M efficiently in step 3, D's tape contains M's current state, symbol read, δ and counter close to each other:
- Split D's single tape into 3 tracks (cells with numbers $\equiv_3 0, 1, 2$, resp.).
  - Track 0 simulates M's tape.
  - Track 1 holds M's state and transition function δ.
  - Track 2 holds the counter.
- When M's head moves on track 0, D shifts contents of tracks 1,2 accordingly (constant time overhead for tracks 0, 1, $O(\log t(n))$ for track 2).
- If M runs in time $r(n)$, then D can simulate M in $O(r(n))$ time (before multiplying $\log t(n)$ to D's running time).

$L(\text{D})$ is computed in $O(\frac{t(n)}{\log t(n)} \cdot \log t(n)) = O(t(n))$ time.

Claim: $L(\mathrm{D})$ is not decidable in $o(\frac{t(n)}{\log t(n)})$ time.

- Let M run in time $r(n)$ with $r(n) = o(\frac{t(n)}{\log t(n)})$.
- In step 3, depending on M's tape alphabet, several cells of D may be needed to encode a single cell of M. Hence D needs $cr(n)$ time to simulate M (without the time counter).
- $\exists N\ \forall n \geq N: \ cr(n) < \frac{t(n)}{\log t(n)}$ and D's simulation runs to completion if M's input has length $\geq N$.
- Then $x := 0^N \sharp(\mathrm{M})$ is accepted by M iff x is rejected by D.
- Hence $L(\mathrm{D}) \neq L(\mathrm{M})$.

□

Consequences of the Time Hierarchy Theorem

Corollary

$\mathrm{DTIME}(n^k) \neq \mathrm{DTIME}(n^\ell)$ if $0 \leq k < \ell$.

Corollary

$\mathrm{P} \neq \mathrm{EXPTIME}$

Proof.

HW □