CS 710: Complexity Theory Date: Jan. 30th, 2024

Lecture 3: Hierarchy Theorems

Instructor: Jin-Yi Cai Scribe: Hao Lin

1 Outline

Prior lecture covers several related topics:

- a 2-tape DTM simulation for any K-tape DTM in $O(T(n) \log T(n))$ time
- definition of complexity classes: P, NP, DTIME, NTIME, DSPACE, NSPACE

In this lecture, we will discuss

- does there exist new problems if we make our complexity classes a little harder: Hierarchy Theorem, especially deterministic versions and non-deterministic space version
- what is the relationship between deterministic and non-deterministic complexity classes, space and time complexity classes: notably Savitch's Theorem
- can we find any completeness in space complexity by reducing specific space-bounded complexity class to a specific problem: determinisitic space reduction

2 Deterministic Hierarchy Theorem

Although we have defined many complexity classes such as P, NP, NSPACE$[\log n]$, we haven't investigated their relationship much yet. There exists many questions yet to be solved, for instance, as a constant will not influence the set size of complexity classes (e.g. DTIME$[f(n)] =$ DTIME$[c \cdot f(n)], \forall c > 0$), how much should we scale the function f such that we can have a larger complexity class? Or, even more fundamental, is that guaranteed to have such a new larger complexity class? The existence of such a hierarchy in complexity classes should be as of similar interesting as the existence of a Turing machine (or say an algorithm according to Turing-Church Thesis) for any problem.

The answer is yes to the latter problem, which is formulated as the Hierarchy Theorem.

Theorem 1. (Deterministic Space Hierarchy Theorem) If $S_2(n)$ is fully space-constructible function, and $S_2(n) \geq \log n$, and

$$\lim_{n\to\infty} \frac{S_1(n)}{S_2(n)} = 0$$

then DSPACE$[S_1(n)] \subsetneqq$ DSPACE$[S_2(n)]$.

Proof. This is a weaker version which is proved by diagonalization method, with loss of minor technical details (the complete theorem relax the condition that fully space-constructible can be replaced by (partially) space-constructible, and the lower bound for $S_2(n)$ is unnecessary). The rows are the enumeration of TM M_i; the columns are the input s_j.

First, we construct a TM $\hat{M_i}$ which simulates the TM M_i where the space it uses for every input of length n is $S_1(n)$. As we know the simulation will not only add a constant factor to the space complexity, so $\hat{M_i}$ uses $O(S_1(n)) = o(S_2(n))$ spaces. Then, to use exactly $S_2(n)$ spaces, it just marking the rest cells on the tape, so the language $\hat{L}$ simulated by $\hat{M_i}$ follows $\hat{L} \in \mathrm{DSPACE}[S_2(n)]$.

Second, we ask all the $\hat{M_i}$ acting on input x doing the opposite: $\hat{M_i}$ rejects x if M_i accepts x, and vice versa.

Finally, according to the diagonalization method, we know there exists an $\hat{M_i}$ whose simulated language $\hat{L}$ does not belong to the $\mathrm{DSPACE}[S_1(n)]$, so $\hat{L} \in \mathrm{DSPACE}[S_2(n)] - \mathrm{DSPACE}[S_1(n)]$. To be more specific, we can find such $\hat{M}$ doing the opposite as mentioned on input s_i when simulating M_i, so we cannot find any M_k of $S_1(n)$ space to have the same behavior as $\hat{M}$ but has $S_2(n)$ space usage. □

Theorem 2. (Deterministic Time Hierarchy Theorem) If $T_1(n), T_2(n)$ are fully time-constructible functions, and $T_1(n) \geq n$, $T_2(n) \geq n$, and

$$\lim_{n\to\infty} \frac{T_1(n) \cdot \log T_1(n)}{T_2(n)} = 0$$

then $\mathrm{DTIME}[T_1(n)] \subsetneq \mathrm{DTIME}[T_2(n)]$.

Proof. For its counterpart, the time hierarchy, the proof is similar except for the major difference is the first stage, the simulation stage, takes a higher complexity order, could be quadratic if simulated by 1-tape DTM. Luckily, we have the 2-tape DTM simulation with $O(T \log T)$, so we have a denser hierarchy, though sparser than the space hierarchy. □

3 Savitch's Theorem & $\mathrm{NSPACE}\left[\log n\right]$ "is" GAP

Proposition 1. $\forall c > 0$,

$$\begin{aligned} \mathrm{DTIME}[T(n)] &\subseteq \mathrm{NTIME}[T(n)] \subseteq \mathrm{DSPACE}[T(n)] \\ \mathrm{NSPACE}[S(n)] &\subseteq \mathrm{DTIME}[c^{S(n)}] \\ \mathrm{NTIME}[T(n)] &\subseteq \mathrm{DTIME}[c^{T(n)}] \end{aligned}$$

Here comes the Savitch's Theorem, a non-trivial but important missing puzzle that bridges the space complexity classes between deterministic and non-deterministic circumstances, with a quadratic more space for DTM simulation.

Definition 1. Graph accessibility problem (GAP) is to test whether there is a path from s to t, given a directed graph $G = (V, E)$ over n nodes.

Lemma 1. GAP can be decided in $O(\log^2 n)$ space.

Proof. We design a recursive algorithm to test given a node w as the potential middle point of the path from s to t, can w reach both points within $[t/2]$ steps. If after iterating of all nodes w, we cannot find one, it leads to a rejection of the problem; otherwise, we can claim we find a path.

The problem is inherently recursive by dividing it into two sub-problems of half size. Specifically, the algorithm named $\mathrm{REACH}(u, v, t)$ calls $\mathrm{REACH}(u, x, t/2)$ and $\mathrm{REACH}(x, v, t/2)$ for every node x. Each call takes $O(\log n)$ space since it only needs to record several variables u, v, t, x, each is written into $\log n$ bits in a binary setting. And the depth of the whole recursive call is $O(\log n)$, so in total we need to spend $O(\log^2 n)$ space. □

Theorem 3. (Savitch's Theorem) If $S(n) \geq \log n$ and is fully space-constructible, then

$$\text{NSPACE}[S(n)] \subseteq \text{DSPACE}[S^2(n)]$$

Proof. The proof is done by translating the general problem into a GAP.

Let M be an NDTM uses $O(S(n))$ space. Our goal is to simulate it with a DTM M' using $O(S^2(n))$ space.

We establish a GAP that V is the configuration graph of M on an input x, which has $2^{O(S(n))}$ nodes, E is the transition from one configuration to the next configuration (defined by the transition relation of M), and the output is whether any of the final configuration ($q \in q_terminal$) is reachable from the the initial configuration.

This GAP can be solved deterministically in $O(\log^2(2^{O(S(n))}) = O(S^2(n))$. □

Definition 2. A language B is NL-*complete* if $B \in$ NL, and for any language $A \in$ NL, there is a deterministic Turing machine M that uses $O(\log n)$ space such that, for any x, $x \in A \iff M(x) \in B$.

Definition 3. We call this reduction from A to B a logspace reduction.

Corollary 1. GAP is NL-*complete*.

Proof. From the proof of Savitch's Theorem, we have constructed a GAP for any language in NL. □

This shows that we can treat NL as a GAP without loss of generality.

4 Non-deterministic Space Hierarchy Theorem

As for NDTM, we are able to come up with a much denser hierarchy for space, compared to DTM.

Lemma 2. (Translational Lemma) Let $S_1(n)$, $S_2(n)$, and $f(n)$ be fully space-constructible, and $S_2(n) \geq n$, $f(n) \geq n$. If $\text{NSPACE}[S_1(n)] \subseteq \text{NSPACE}[S_2(n)]$, then

$$\text{NSPACE}[f(S_1(n))] \subseteq \text{NSPACE}[f(S_2(n))]$$

Proof. Let $L_1 \in \text{NSPACE}[S_1(f(n))]$ defined by M_1.

For any input x that is accepted by M_1 in space $S_1(f(|x|))$, we pad the input with # to be $x\#\# \ldots \#\#$, where there are $f(|x|) - |x|$ many padding character # (which is of in total $f(|x|)$ length), we define L_2 to be all these padded strings. Therefore, $L_2 \in \text{NSPACE}[S_1]$.

Since $\text{NSPACE}[S_1(n)] \subseteq \text{NSPACE}[S_2(n)]$, we have $L_2 \in \text{NSPACE}[S_2]$.

Apply $f(n)$ as the parameter to both results, we have $L_1 \in \text{NSPACE}[S_2(f(n))]$. □

Theorem 4. If $\epsilon > 0$ and $r \geq 0$, then

$$\text{NSPACE}[n^r] \subsetneq \text{NSPACE}[n^{r+\epsilon}]$$

Proof. Considering the dense nature of rational number, we can find s and t such that $r \leq s/t < (s+1)/t \leq r + \epsilon$, so we only need to prove its rational version

$$\text{NSPACE}[n^{s/t}] \subsetneqq \text{NSPACE}[n^(s+1)/t]$$

We prove by contradiction.

Suppose $\text{NSPACE}[n^(s+1)/t] \subseteq \text{NSPACE}[n^{s/t}]$, then with translational lemma, take $f(n) = n^{(s+i)t}$, we have

$$\text{NSPACE}[n^{(s+1)(s+i)}] \subseteq \text{NSPACE}[n^{s(s+i)}], \forall i = 0, 1, \ldots, s$$

Also, for $i \geq 1$, $s(s+i) \leq (s+1)(s+i-1)$, we have

$$\text{NSPACE}[n^{s(s+i)}] \subseteq \text{NSPACE}[n^{(s+1)(s+i-1)}]$$

Taking these two results alternatively, we have

$$\begin{aligned} \text{NSPACE}[n^{(s+1)(2s)}] &\subseteq \text{NSPACE}[n^{s(2s)}] \\ &\subseteq \text{NSPACE}[n^{(s+1)(2s-1)}] \subseteq \text{NSPACE}[n^{s(2s-1)}] \\ &\subseteq \ldots \text{NSPACE}[n^{s^2}] \end{aligned}$$

However, by Savitch's Theorem, $\text{NSPACE}[n^{s^2}] \subseteq \text{DSPACE}[n^{2s^2}]$, and we also have $\text{DSPACE}[n^{2s^2}] \subsetneqq \text{DSPACE}[n^{2s^2+2s}]$, and $\text{DSPACE}[n^{2s^2+2s}] \subseteq \text{NSPACE}[n^{2s^2+2s}]$.

So, we have a contradiction that $\text{NSPACE}[n^{2s^2+2s}] \subsetneqq \text{NSPACE}[n^{2s^2+2s}]$. $\square$