Lecture 10

Gidon Rosalki

2025-06-08

1 The hierarchy theorems

Theorem 1 (Hierarchy theorem of time)**.** *For every function f that is calculable in $O(f)$, it holds that:*

$$Time(o(f(n))) \subsetneq Time(f(n)\log(n))$$

(Yes, that is meant to be a little o)

Theorem 2 (Hierarchy theorem of space)**.** *For every function f that is calculable in space $O(f)$ it holds that*

$$Space(o(f(n))) \subsetneq Space(f(n))$$

(Yes, that is meant to be a little o)

1.1 Hierarchy theorem in time

We will return to the proof that shows that $R \subsetneq RE$, and if we update it with limitation on time / space, then we get the Hierarchy theorems above. Reminder:

$$A_{TM} = \{\langle M\rangle, w : M(w) = q_{\text{acc}}\}$$

We will assume by contradiction that there exists a DTM H that decides A_{TM}. This is to say

$$H(\langle M\rangle, w) = q_{\text{acc}} \Leftrightarrow M(w) = q_{\text{acc}}$$

We will define $D(\langle M\rangle) = H(\langle M\rangle, \langle M\rangle)$, and $\overline{D}(\langle M\rangle) = \overline{D(\langle M\rangle)}$ (which is to say, swapping between $q_{\text{acc}}, q_{\text{rej}}$. If

$$\begin{aligned}\overline{D}\left(\left\langle\overline{D}\right\rangle\right) = q_{\text{acc}} &\implies D\left(\left\langle\overline{D}\right\rangle\right) = q_{\text{rej}}\\ &\implies H\left(\left\langle\overline{D}\right\rangle, \left\langle\overline{D}\right\rangle\right) = q_{\text{rej}}\\ &\implies \overline{D}\left(\overline{D}\right) = q_{\text{rej}}\end{aligned}$$

and if

$$\begin{aligned}\overline{D}\left(\left\langle\overline{D}\right\rangle\right) = q_{\text{rej}} &\implies D\left(\left\langle\overline{D}\right\rangle\right) = q_{\text{acc}}\\ &\implies H\left(\left\langle\overline{D}\right\rangle, \left\langle\overline{D}\right\rangle\right) = q_{\text{acc}}\\ &\implies \overline{D}\left(\overline{D}\right) = q_{\text{acc}}\end{aligned}$$

so in both cases, we arrive at a contradiction.

Let us define a language:

$$A_f = \{(\langle M\rangle, w) : M(w) = q_{\text{acc}} \wedge \text{Running } M \text{ on } w \text{ finishes in at most } f(|w|) \text{ steps}\}$$

We may decide the language A_f by using a UTM (in how much time?) We will show, with a similar proof to that of A_{TM} that we may not decide $A)f$ in time $o(f(n))$.
Let us assume by contradiction that there is a TM H_f that decides A_f in $o(f(n))$ steps. We will use H_f to build D:

$$D(\langle M\rangle) = H_f(\langle M\rangle, \langle M\rangle)$$

We will then use D to build $\overline{D}$:

$$\overline{D}(\langle M\rangle) = \overline{D(\langle M\rangle)}$$

Which is to say, a swapping of q_{acc} and q_{rej}. So, what does $\overline{D}\left(\left\langle\overline{D}\right\rangle\right)$ return?
If $\overline{D}\left(\left\langle\overline{D}\right\rangle\right) = q_{\text{acc}}$, then $D\left(\left\langle\overline{D}\right\rangle\right) = q_{\text{rej}} \implies H_f\left(\left\langle\overline{D}\right\rangle, \left\langle\overline{D}\right\rangle\right) = q_{\text{rej}}$, and so $\overline{D}\left(\left\langle\overline{D}\right\rangle\right) = q_{\text{rej}}$, which is a contradiction.
(Note: The runtime of H_f is $o(f(n))$, and therefore so is D, and $\overline{D}$)
If $\overline{D}\left(\left\langle\overline{D}\right\rangle\right) = q_{\text{rej}}$, then $D\left(\left\langle\overline{D}\right\rangle\right) = q_{\text{acc}} \implies H_f\left(\left\langle\overline{D}\right\rangle, \left\langle\overline{D}\right\rangle\right) = q_{\text{acc}}$, and so we get that $\overline{D}\left(\left\langle\overline{D}\right\rangle\right) = q_{\text{acc}}$, which is once more a contradiction.

We have shown here that there exists a language $A_f \notin \text{Time}(o(f(n)))$. It is clear that A_f is decidable, the question is in what runtime. We may decide A_f as follows: Initialise a timer for $f(|w|)$, run the UTM U: $U(\langle M\rangle, w)$, and after simulating each step, we reduce the value of the timer by 1. If the timer reaches 0, we will stop and return q_{rej}, and if before the timer reaches 0, the simulation of M reaches q_{acc}, or q_{rej}, then we will return accordingly. This machine is clearly a deciding machine of A_f, but how much time does it take? We need to consider initialising the timer, and reducing its value every step. There is an implementation of a UTM U, that simulates the run of $f(n)$ steps, in $O(f(n)\log(f(n)))$ steps of U. We will require that the function f is calculable in time $O(f(n))$. This is to say, given 1^n, we may compute $1^{f(n)}$ in $O(fbrn)$.
Conclusion: For every function f, that is computable in $O(f)$, it is true that

$$\text{Time}o(f(n)) \subsetneq \text{Time}(f(n\log(f(n))))$$

Example: n^2 is calculable in $O(n^2)$, and also $n^2 = o(n^{2.1})$, and also $n^2 \cdot \log(n^2) = O(n^3)$, and so $\text{Time}(n^2) \subsetneq \text{Time}(n^3)$.

From all this, we may conclude that $P \subsetneq Exp$, for example there is a language $L \in \text{Time}(2^n)$, and $L \notin \text{Time}(n^k)$ for any constant k.

1.2 Hierarchy theorem of space

For every function f that is computable in space $O(f)$ it is true that

$$\text{Space}(o(f(n))) \subsetneq \text{Space}(f(n))$$

The proof is almost identical to the proof for time, however we will simply replacement the time limitation at every point of the proof, with a limitation of space. The theorem is stronger for space (tighter separation), since there exists a UTM Y that simulates $M(w)$ in space $O(f(n))$ where $f(n)$ is the space limitation of the run of M.
We can limit the space, similarly to the timer, where we begin by adding a symbol to every cell in the tape that we are allowed to use, and if the simulation tries to go beyond these cells, we reject.

2 Savitch theorem

$$NSpace(f(n)) \subseteq Space(f^2(n))$$

For every function $f(n)$ that is computable in space $O(f(n))$
From this we may conclude that

$$NPSPACE = PSPACE$$

Reminder:

$$PSPACE = \bigcup_{k=1}^{\infty} SPACE(n^k)$$

$$NPSPACE = \bigcup_{k=1}^{\infty} NSPACE(n^k)$$

Proof of conclusion: $PSPACE \subseteq NPSPACE$ - this is trivial, since every DTM is a special case of an NTM with the same space limitation. We will show the second direction: Let there be $L \in NPSPACE$, which is to say, there exists k, some constant, such that $L \in NPSPACE(n^k)$. By the Savitch theorem,

$$L \in SPACE\left(\left(n^k\right)^2\right) = SPACE(n^{2k}) \subseteq PSPACE$$

2.1 Savitch proof

For a TM M, and an input w, we will define the configuration graph $G_{M,w}$. The nodes of $G_{M,w}$ will be all the possible configurations. There exists an directed edge from u to v **if and only if** the configuration represented by v is the configuration that follows the configuration that is represented by u. For a DTM, for every configuration, there is a single following configuration, and therefore for every node, there is a single edge that leaves. However, in an NTM, there are many configurations that follow a given configuration, and therefore for every node, there may be more than one edge that leaves it.
We will define 2 nodes in the graph: The node c_0 which is the starting configuration: $c_0 = q_0 w$. Additionally, the node c_{acc} which represents the accepting configuration $c_{\text{acc}} = q_{\text{acc}}$.
Reminder: We have shown in the proof of Cook's theorem, that w.l.o.g. there is a single accepting configuration. Note that the change to M in order so that there is a single accepting configuration does not increase the space complexity.
We will note that $w \in L \Leftrightarrow$ there is an accepting run of $M(w) \Leftrightarrow$ in the graph $G_{M,w}$ there is a directed path from c_0 to c_{acc}.

So, how many nodes are there in $G_{M,w}$, or rather, how many possible configurations are there?

$$\text{Num configurations} = \text{Number of states} \times \text{Head locations} \times \text{Tape contents} = |Q| \cdot O(f(n)) \cdot |\Gamma|$$

It is clear that we may decide L through a DTM as follows: We will create $G_{M,w}$. Run on it BFS, or DFS, starting from c_0. We will return q_{acc} **if and only if** we arrive at c_{acc}.
Correctness is trivial.
Space complexity is $\geq$ the number of nodes in $G_{M,w}$, which is exponential in $f(n)$.

For a language $L \in NSPACE(f(n))$, there exists an NTM M, which runs in space $O(f(n))$. We want to show that there exists a DTM M', that decides L, in $O\left(f^2(n)\right)$ space.

The procedure $Reach(u, v, t)$ answers the question: Does there exist a path in $G_{M,w}$ that starts at u, and finishes at v, with length $\leq t$? We will show a DTM that computes $Reach$.

Reach 1

Input: u, v, t
Output: *bool*

if $u = v$ **then**
  return T
end if
if v is a successor of u **then**
  return T
end if
if $t \leq 1$ **then**
  return F
end if
for $m \in V(G_{M,w})$ **do**
  $q_1 \leftarrow Reach\left(u, m, \left\lceil \frac{t}{2} \right\rceil\right)$
  $q_2 \leftarrow Reach\left(u, m, \left\lfloor \frac{t}{2} \right\rfloor\right)$
  if $q_1 \wedge q_2$ **then**
    return T
  end if
end for
return F

Note: We will initialise the run with $u = c_0$ and $v = c_{\text{acc}}$, and $t = \#conf$. The correctness is trivial, and we will analyse the runtime: For every recursive call we will create a new 3-tuple, but 2 calls of the same depth are used for the same space. We want to analyse the space complexity in M'. A check of if $u = v$ does not require any additional space. A check of if v follows u does not require any additional space. How much space is required for a single 3-tuple?

$$\begin{aligned} u &= \log\left(|Q| \cdot O(f(n)) \cdot |\Gamma|^{O(f(n))}\right) \\ &= \log|Q| + O(\log(f(n))) + O(f(n)) \cdot \log|\Gamma| \\ &= O(f(n)) \end{aligned}$$

Similarly, v requires $O(f(n))$, and t requires $O(f(n))$. So, in total for the whole 3-tuple: $O(f(n))$. The maximum number of 3-tuples is the maximum depth of the recursion, which is in total $O(\log(\#conf))$, which is to say $O(f(n))$, so in total, our space complexity is $O\left(f^2(n)\right)$.

So the space complexity is sublinear: Examples $L = Space(\log(n))$, and $NL = NSpace(\log(n))$

We will create a TM that is suitable for discussing sublinear space. We will use a machine with 3 tapes: Input, work, and output, where input and work are read only, and output is write only. The head of the output tape can, at every run step, either print a letter, or not. If it prints, it moves one step to the right, and does not return. Let us define δ:

$$\delta : Q \times \Sigma \times \Gamma \to Q \times \Gamma \times \{\Sigma \cup \emptyset\}$$

The run space is measured purely from the work tape, and so the output tape cannot return to the left, since this would increase the work space.

Consider the example: Language $A \in Space(\log(n))$ L **if and only if** There exists a DTM M that decides A, and in the space $O(\log(n))$. This is to say that M is a machine of 2 tapes, input and work. The space limitation on the work tape is $O(\log(n))$.
Similarly, $B \in NL$ if there exists an NTM that decides B, and in space that is logarithmic, which is to say, 2 tapes, input, and work, with a size limitation on the work tape of $O(\log(n))$.

Theorem 3 ($L \subseteq P$)**.** *This is to say that $A \in L \implies A \in P$, or if there is a DTM that is run in logarithmic space, then there is a DTM that runs in polynomial time.*

Proof . Let there be M a DTM that decides A in a logarithmic amount of space. How many configurations does this machine have? Number of states, times the location of the input head, times the location of the work head, times the

contents of the work tape:

$$\begin{aligned}|Q| \cdot n \cdot O(\log(n)) \cdot |\Gamma|^{O(\log(n))} &= |Q| \cdot n \cdot O(\log(n)) \cdot 2^{\log_2(|\Gamma|) \cdot O(\log_2(n))} \\ &= O(n^c)\end{aligned}$$

Where c is some constant. Overall, the number of configurations is $O\left(n^d\right)$, where d is some constant. It is clear that M does not return to the same configuration in a run, since it if did, it would be in an infinite loop.
Conclusion, the runtime of M on $w \leq$ the number of possible configurations, which is $O\left(n^d\right)$, which is to say, a polynomial runtime, and so $A \in P$ □

Theorem 4. *If M calculates the function f, and M is a DTM, that runs in space $O(\log(n))$, then the length of the input $|f(w)|$ is bound from above by a polynomial.*

Proof . The proof is similar to the previous theorem. The number of configurations is bound by a polynomial, and therefore the number of steps, and the number of prints, are also bound by a polynomial. □