# 6. Space and Time Hierarchies

Goal: Show that more space and more time mean more power.

Proof technique: Diagonalization

↳ Show the existence of a language $L$ (with certain properties, here space requirements) that cannot be decided by a TM drawn from a given set $S = \{M_1, M_2, \dots\}$ (again here defined by space requirements).

↳ Start with some language $L_0$ (having the property).

Then for $i = 1, 2, \dots$ do

change $L_{i-1}$ to $L_i$ so that

↳ none of $M_1, \dots, M_i$ decides $L_i$

↳ $L_i$ still has the property.

The language $L$ is defined as the limit of this construction.

Warm-up on diagonalization:

Lemma: There are undecidable languages.

Proof (1-line version):

The set of languages over $\{0,1\}$ is $\mathcal{P}(\{0,1\}^*)$ and hence uncountable.

The set of TMs is countable. $\square$

Proof (via diagonalization):

Consider an enumeration $x_1, x_2, \dots$ of all binary strings and an enumeration $M_1, M_2, \dots$ of all TMs over $\{0,1\}$.

Define $L := \{x_i \in \{0,1\}^* \mid M_i(x_i) = 0\}$.

Illustration:

$$
\begin{array}{c|cccccc}
 & x_1 & x_2 & x_3 & x_4 & \cdots & \\
\hline
M_1 & 1 & & & & & \\
M_2 & & 0 & & & & \\
M_3 & & & 0 & & & \text{defines } L. \\
M_4 & & & & 1 & & \\
\vdots & & & & & \ddots &
\end{array}
$$

Towards a contradiction, assume some machine $M$ decides $L$.

Let $i \in \mathbb{N}$ be such that $M = M_i$.

Consider now $x_i \in \{0,1\}^*$.

If $M(x_i) = 1$, then $x_i \notin L$, so $M$ is wrong.

If $M(x_i) = 0$, then $x_i \in L$, so $M$ is also wrong. $\square$

## 6.1 Universal Turing Machines

<u>Goal</u>: • For diagonalization, we have to (1) encode and (2) simulate Turing machines.

- For (1), we need a Gödel numbering of Turing machines. For (2), we need a universal TM.
- Note that the universal TM should be an <u>efficient</u> simulator, i.e., obey time and space bounds.

The size of the TM encoding will usually be a constant ($M$ fixed, input varies), so there is no point to be space efficient.

Moreover, the encoding should be easy to simulate.

Definition (Unary Encoding of DTMs into $\{0,1\}^*$):

Consider a $k$-tape DTM $M = (Q, \Sigma, \Gamma, \triangleright, \sqcup, \delta, q_{init}, q_{accept}, q_{reject})$

Let $Q = \{\underset{\overset{\shortparallel}{q_1}}{q_{init}}, \underset{\overset{\shortparallel}{q_2}}{q_{accept}}, \underset{\overset{\shortparallel}{q_3}}{q_{reject}}, \dots, q_{|Q|}\}$, $\Gamma = \{\underset{\overset{\shortparallel}{\gamma_1}}{\sigma_1}, \dots, \underset{\overset{\shortparallel}{\gamma_{|\Sigma|}}}{\sigma_{|\Sigma|}}, \underset{\overset{\shortparallel}{\gamma_{|\Sigma|+1}}}{\triangleright}, \overset{\overset{\gamma_{|\Sigma|+2}}{\shortparallel}}{\sqcup}, \dots, \gamma_{|\Gamma|}\}$

We encode the transitions

$$\delta(q_i, \gamma_{i1}, \dots, \gamma_{ik}) = (q_j, \gamma_{j1}, \dots, \gamma_{jk}, d_1, \dots, d_k)$$

as

$$\underbrace{0^i}_{\text{state}} 1 \underbrace{0^{i_1}}_{\substack{\text{letter}\\ \text{tape 1}}} 1 \dots 1 \underbrace{0^{i_k}}_{\substack{\text{letter}\\ \text{tape } k}} 1 \underbrace{0^j}_{\text{state}} 1 \underbrace{0^{j_1}}_{\substack{\text{new}\\ \text{letter}\\ \text{tape 1}}} 1 \dots 1 \underbrace{0^{j_k}}_{\substack{\text{new}\\ \text{letter}\\ \text{tape } k}} 1 \underbrace{0^{2+d_1}}_{\substack{\text{direction}\\ \text{tape 1}}} 1 \dots 1 \underbrace{0^{2+d_k}}_{\substack{\text{direction}\\ \text{tape } k}}.$$

Here, we assume the directions are given as $d$ from

- $-1 :=$ left
- $0 :=$ stay
- $1 :=$ right.

The <u>encoding of $M$</u> is

$$enc(M) := \underbrace{0^k}_{\text{tapes}} 11 \underbrace{0^{|Q|}}_{\text{states}} 11\, 0^{|\Gamma|} 11 \underbrace{0^{|\Sigma|}}_{\substack{\text{First } |\Sigma|\\ \text{Elements of } \Gamma\\ \text{are letters}\\ \text{from the input}\\ \text{alphabet}}} 11 \underbrace{enc(\delta)}_{\substack{\text{Transitions as above,}\\ \text{separated by 11.}}}$$

<u>Theorem:</u>

There is a 1-tape DTM $U$ so that $U$ on input $\langle e, x\rangle$ computes $E(x)$, where $e$ is the above encoding of the DTM $E$.

If $E$ uses space $s(n)$, then $U$ uses space $O(|e| \cdot s(n))$.

For each step of $E$, $U$ does $O(|e|^2 \cdot s(n))$ steps.

Here, $\langle e,x \rangle$ is the encoding of $E$
plus the encoding of the input $x = x_1, \dots, x_n \in \Sigma_E^*$.
It takes the form

$$\langle e,x \rangle := enc(E)\,1111\,0^{x_1}1\,0^{x_2}1 \dots 1\,0^{x_n}.$$

The machine $U$ is called <u>universal</u> for the class of DTMs.
$U$ can be modified to an NTM that is universal for the class of NTMs.
As an intuition, universal machines can be understood
as assembly interpreters written in assembly.

<u>Proof idea:</u>

- Assume the given DTM $E$ has $k$ tapes.
  $U$ stores them on one tape with the tape reduction trick,
  i.e., our new letters have $2k$ tracks.
- There is a problem:
  ↳ there is no bound on the size of the work alphabet $\Gamma_E$ of $E$
  ↳ nevertheless, we have to fix the work alphabet $\Gamma_U$ of $U$.

  The solution is to store a symbol

$$\begin{pmatrix} \gamma_1 \\ m_1 \\ \vdots \\ \gamma_k \\ m_k \end{pmatrix} \in (\Gamma_E \times \{*, -\})^k$$

  as a string of length $|\Gamma_E|$ of the form

$$\begin{pmatrix} f(\gamma_1) \\ g(m_1) \\ \vdots \\ f(\gamma_k) \\ g(m_k) \end{pmatrix} \in \left(\{0,1\}^{2k}\right)^{|\Gamma_E|} \quad \text{with} \quad \begin{aligned} f(\gamma_h) &:= 0^h 1^{|\Gamma_E|-h} \\ g(*) &:= 1^{|\Gamma_E|} \\ g(-) &:= 0^{|\Gamma_E|} \end{aligned}$$

- The current state $q_h$ of $E$ is stored as a string $0^h 1^{|Q|-h}$
  of length $|Q|$.

We place this string for the state at the beginning of the work tape of $\mathcal{U}$, followed by the above encoding of the work tape of $E$.

- The simulation of one transition is as follows.
  - ↳ $\mathcal{U}$ goes through the transition function to find the first entry, where the state matches the current state of $E$. (To this end, $\mathcal{U}$ will compare the 0s after the 11 for enc($\delta$) symbol by symbol to see whether they match the current state stored at the beginning of the work tape.)
  - ↳ If the state does not match, $\mathcal{U}$ goes to the next entry of the transition function.
  - ↳ If a transition for the state is found, $\mathcal{U}$ will go over the work tape to check whether the current symbols match what the transition expects.
    - → If the symbols do not match, $\mathcal{U}$ finds the next transition.
    - → If the symbols do match, $\mathcal{U}$ moves back over the tape and performs the required changes.

- The time requirement to simulate a single step of $E$ is

$$O(|e| \cdot 2 \cdot |e| s(n)) = O(|e|^2 s(n)).$$

There are $\leq |e|$ transitions.
For each we may have to scan the whole tape (back and forth).
(On the way back, we may have to perform changes.)
Since the tape has a length of $|e| \cdot s(n)$,
we arrive at $O(\underbrace{|e|}_{\text{each transition}} \cdot \underbrace{2|e|s(n)}_{\text{scan tape back and forth}})$ steps.

$\square$

## 6.2 Deterministic Space Hierarchy

Theorem (Deterministic Space Hierarchy):

Let $s_2(n) \geq \log n$ be space constructible

and let $s_1 = o(s_2)$.

Then

$$DSPACE(s_1) \subsetneq DSPACE(s_2).$$

In particular, $L \subsetneq PSPACE$.

Proof:

Let $U$ be the universal TM from the previous theorem.

We construct a TM $M$ that

↳ is $s_2$-space bounded and

↳ so that $L(M) \notin DSPACE(s_1)$.

$M$ works as follows:

Input: $y \in \{0,1\}^*$, interpreted as $\langle e, x \rangle$.

Output: $0$ if the TM $E$ encoded by $e$ accepts $y$

$1$ otherwise.

Begin:

1. Mark $s_2(|y|)$ cells on the tape
2. Let $y = \langle e, x \rangle$.
   Check whether $e$ is a valid encoding of a DTM $E$.
   Can be done in $\log |y|$ space.
3. $M$ now simulates $E$ on input $y$ (not $x$).
   To this end, $M$ behaves like $U$.
4. On an extra tape, $M$ counts the steps of $U$,
   using a ternary counter with $s_2(|y|)$ digits.
5. If during the simulation, $U$ leaves the marked space,
   $M$ rejects.

6. If $E$ halts, $M$ halts. If $E$ accepts, $M$ rejects.
If $E$ rejects, $M$ accepts.

7. If $E$ makes more than $3^{s_2(|y|)}$ steps, $M$ accepts.

End.

To show that $L(M)$

$$L(M) = \{ \langle e, x \rangle \mid E \text{ does not accept } \langle e, x \rangle \text{ in space} \leq s_2(\langle e, x \rangle) \} \notin DSPACE(s_1),$$

we proceed by contradiction
and assume this was the case.
Let $N$ be an $s_1$-space bounded and dated DTM
with
$$L(N) = L(M).$$
It is sufficient to consider $N$ 1-tape (with extra input tape).
Let $e$ be an encoding of $N$
and let $y = \langle e, x \rangle$ for $x$ sufficiently long.

1) $y \in L(M)$.

If $M$ accepts $y$,
then either the simulation of $N$ terminated
or $N$ makes more than $3^{s_2(|y|)}$ steps.

→ If $N$ terminated and $M$ accepted $y$,
then $N$ rejected $y$,
a contradiction to the assumption $L(M) = L(N)$.

→ In the latter case, $N$ cannot make more than

$$c^{s_1(|y|)} (s_1(|y|) + 2)(|y| + 2)$$

steps (otherwise, $N$ would enter an infinite loop).

Thus, if

$$3^{s_2(|y|)} > c^{s_1(|y|)} \cdot (s_1(|y|)+2)(|y|+2)$$

$$\iff \underbrace{\log 3}_{>1} \cdot s_2(|y|) > \log c \cdot s_1(|y|) + \log(s_1(|y|)+2) + \log(|y|+2),$$

we can enforce termination and derive a contradiction, too.

For long enough $x$, the inequality holds.

2) $y \notin L(M)$.

If $M$ rejects $y$.

$M$ ran out of space or $N$ terminated.

$\rightarrow$ The termination case is again easy:

since $y \notin L(M)$, $N$ accepts $y$.

$\rightarrow$ We show that the first case does not happen.

Since $N$ is $s_1$-space bounded,

the simulation via $U$ needs $|e| \cdot s_1(|y|)$ space.

But

$$|e| \cdot s_1(|y|) \leq s_2(|y|)$$

for sufficiently large $x$.

Note that $M$ is $s_2$-space bounded. $\square$

## 6.3 Further Hierarchy Results

For time complexity, the separation result will not be as nice.

Why? The universal TM is slower than the given TM
by a quadratic factor.

**Lemma (Deterministic Time Hierarchy):**
Let $t_2$ be time constructible and $t_1^2 = o(t_2)$.
Then

$$DTIME(t_1) \subsetneq DTIME(t_2).$$

In particular, $PTIME \subsetneq EXP$.

Hennie & Stearns showed a more efficient universal TM construction
that allows us to strengthen the theorem.

**Theorem (Hennie & Stearns '66):**
Let $t_2$ be time constructible and $t_1 \cdot \log t_1 = o(t_2)$.
Then

$$DTIME(t_1) \subsetneq DTIME(t_2).$$

For non-deterministic space, we can combine
the deterministic space hierarchy result
with Savitch's theorem.

**Theorem (Non-Deterministic Space Hierarchy):**
Let $s_1(n) \geq \log n$ be space constructible
and let $s_1 = o(s_2)$.
Then

$$NSPACE(s_1(n)) \overset{\text{Savitch}}{\subseteq} DSPACE(s_1(n)^2) \overset{\text{DSH}}{\subsetneq} DSPACE(s_2(n)^2) \subseteq NSPACE(s_2(n)^2)$$

In particular, $NL \subsetneq PSPACE$.