Lesson 10. Basic complexity classes

CSIE 3110 – Formal Languages and Automata Theory

Tony Tan
Department of Computer Science and Information Engineering
College of Electrical Engineering and Computer Science
National Taiwan University

Table of contents

Table of contents

What we will cover in this lesson

From our experience in programming, we just “know” (intuitively) that some (computational) problems are more “difficult” than others.

What we will cover in this lesson

From our experience in programming, we just "know" (intuitively) that some (computational) problems are more "difficult" than others.

Multiplying two numbers is more difficult than adding them.

What we will cover in this lesson

From our experience in programming, we just "know" (intuitively) that some (computational) problems are more "difficult" than others.

Multiplying two numbers is more difficult than adding them.

Sorting a list of numbers is more difficult than finding the maximum element.

What we will cover in this lesson

From our experience in programming, we just "know" (intuitively) that some (computational) problems are more "difficult" than others.

Multiplying two numbers is more difficult than adding them.

Sorting a list of numbers is more difficult than finding the maximum element.

The language HALT is more difficult than HALT_0 (even if both are undecidable).

What we will cover in this lesson

From our experience in programming, we just “know” (intuitively) that some (computational) problems are more “difficult” than others.

Multiplying two numbers is more difficult than adding them.

Sorting a list of numbers is more difficult than finding the maximum element.

The language HALT is more difficult than HALT_0 (even if both are undecidable).

The classification of languages/problems according to their “difficulty” is an important area in computer science.

Classification in this lesson

Classification according to number of steps and cells in the tape needed by Turing machines to decide them.

Classification in this lesson

Classification according to number of steps and cells in the tape needed by Turing machines to decide them.

The classification according to the number of steps is called *the time complexity*.

Classification in this lesson

Classification according to number of steps and cells in the tape needed by Turing machines to decide them.

The classification according to the number of steps is called *the time complexity*.

The classification according to the number of cells is called *the space complexity*.

Classification in this lesson – continued

Classification in this lesson – continued

- The class of problems decidable by polynomial time DTM and NTM, denoted by **P** and **NP**, respectively.

Classification in this lesson – continued

- The class of problems decidable by polynomial time DTM and NTM, denoted by **P** and **NP**, respectively.
- The class of problems decidable by polynomial space DTM and NTM, denoted by **PSPACE** and **NPSPACE**, respectively.

Classification in this lesson – continued

- The class of problems decidable by polynomial time DTM and NTM, denoted by **P** and **NP**, respectively.
- The class of problems decidable by polynomial space DTM and NTM, denoted by **PSPACE** and **NPSPACE**, respectively.
- The class of problems decidable by logarithmic space DTM and NTM, denoted by **L** and **NL**, respectively.

We will also discuss some basic relations between all these classes.

Table of contents

Recall

$\mathbb{N}$ denotes the set of natural numbers $\{0, 1, 2, \ldots\}$.

Recall

$\mathbb{N}$ denotes the set of natural numbers $\{0, 1, 2, \ldots\}$.

Let $f, g : \mathbb{N} \to \mathbb{N}$ be functions.

(Def.) $f(n) = O(g(n))$ means there is $c, n_0 \in \mathbb{N}$ such that for every $n \geqslant n_0$:

$$f(n) \leqslant c \cdot g(n)$$

Recall

$\mathbb{N}$ denotes the set of natural numbers $\{0, 1, 2, \ldots\}$.

Let $f, g : \mathbb{N} \to \mathbb{N}$ be functions.

(Def.) $f(n) = O(g(n))$ means there is $c, n_0 \in \mathbb{N}$ such that for every $n \geqslant n_0$:

$$f(n) \leqslant c \cdot g(n)$$

Recall also that for a word $w \in \Sigma^*$, $|w|$ denotes the length of w.

Polynomial time complexity

(Def.) Let $k \geqslant 1$ be a fixed integer.

A DTM/NTM $\mathcal{M}$ runs in time $O(n^k)$, if:

There is $c, n_0 \in \mathbb{N}$ such that for every word $w \in \Sigma^*$ with $|w| \geqslant n_0$,

every run of $\mathcal{M}$ on w has length $\leqslant c|w|^k$.

Polynomial time complexity

(Def.) Let $k \geqslant 1$ be a fixed integer.

A DTM/NTM $\mathcal{M}$ runs in time $O(n^k)$, if:

There is $c, n_0 \in \mathbb{N}$ such that for every word $w \in \Sigma^*$ with $|w| \geqslant n_0$,

every run of $\mathcal{M}$ on w has length $\leqslant c|w|^k$.

That is, for every run of $\mathcal{M}$ on w with $|w| \geqslant n_0$:

$$C_0 \vdash C_1 \vdash \cdots \vdash C_N \qquad\qquad C_N \text{ can be acc./rej.}$$

we have $N \leqslant c|w|^k$.

Intuitively, each $\vdash$ counts as one step (i.e., each time a transition is applied).

Polynomial time complexity

(Def.) Let $k \geqslant 1$ be a fixed integer.

A DTM/NTM $\mathcal{M}$ runs in time $O(n^k)$, if:

There is $c, n_0 \in \mathbb{N}$ such that for every word $w \in \Sigma^*$ with $|w| \geqslant n_0$,

every run of $\mathcal{M}$ on w has length $\leqslant c|w|^k$.

That is, for every run of $\mathcal{M}$ on w with $|w| \geqslant n_0$:

$$C_0 \vdash C_1 \vdash \cdots \vdash C_N \qquad\qquad C_N \text{ can be acc./rej.}$$

we have $N \leqslant c|w|^k$.

Intuitively, each $\vdash$ counts as one step (i.e., each time a transition is applied).

(Note) The definition is the same for both DTM and NTM.

The only difference is a DTM have only one run on each input word w, whereas an NTM may have many runs.

Polynomial time complexity – continued

(Def.) A *DTM/NTM $\mathcal{M}$ decides/accepts a language L in time $O(n^k)$*, if:

- $\mathcal{M}$ decides L.
- $\mathcal{M}$ runs in time $O(n^k)$.

Polynomial time complexity – continued

(Def.) A *DTM/NTM $\mathcal{M}$ decides/accepts a language L in time $O(n^k)$*, if:

- $\mathcal{M}$ decides L.
- $\mathcal{M}$ runs in time $O(n^k)$.

(Recall) $\mathcal{M}$ decides a language L, if for every word $w \in \Sigma^*$:

$$\mathcal{M} \text{ accepts } w \qquad \text{if and only if} \qquad w \in L$$

The class $\textsc{Dtime}[n^k]$ and P

(Def.) For a fixed integer $k \in \mathbb{N}$:

$$\textsc{Dtime}[n^k] \;:=\; \{L \;|\; \text{there is a } \underline{\text{DTM}}\ \mathcal{M} \text{ that decides } L \text{ in time } O(n^k)\}$$

The class $\textsc{Dtime}[n^k]$ and P

(Def.) For a fixed integer $k \in \mathbb{N}$:

$$\textsc{Dtime}[n^k] \;:=\; \{L \mid \text{there is a } \underline{\text{DTM}}\ \mathcal{M} \text{ that decides } L \text{ in time } O(n^k)\}$$

(Def.)

$$\mathbf{P} \;:=\; \bigcup_{k \geqslant 1} \textsc{Dtime}[n^k]$$

The class $\textsc{Dtime}[n^k]$ and **P**

(Def.) For a fixed integer $k \in \mathbb{N}$:

$$\textsc{Dtime}[n^k] \;:=\; \{L \;|\; \text{there is a } \underline{\text{DTM}}\ \mathcal{M} \text{ that decides } L \text{ in time } O(n^k)\}$$

(Def.)

$$\mathbf{P} \;:=\; \bigcup_{k \geqslant 1} \textsc{Dtime}[n^k]$$

(Note) The class **P** is closed under complement, union and intersection.

The class $\textsc{Ntime}[n^k]$ and NP

(Def.) For a fixed integer $k \in \mathbb{N}$:

$$\textsc{Ntime}[n^k] \ := \ \{L \ | \ \text{there is an } \underline{\text{NTM}} \ \mathcal{M} \text{ that decides } L \text{ in time } O(n^k)\}$$

The class $\textsc{Ntime}[n^k]$ and NP

(Def.) For a fixed integer $k \in \mathbb{N}$:

$$\textsc{Ntime}[n^k] \ := \ \{L \ \mid \ \text{there is an } \underline{\text{NTM}} \ \mathcal{M} \text{ that decides } L \text{ in time } O(n^k)\}$$

(Def.)

$$\mathbf{NP} \ := \ \bigcup_{k \geqslant 1} \textsc{Ntime}[n^k]$$

The class $\textsc{Ntime}[n^k]$ and **NP**

(Def.) For a fixed integer $k \in \mathbb{N}$:

$$\textsc{Ntime}[n^k] \;:=\; \{L \mid \text{there is an } \underline{\text{NTM}}\ \mathcal{M} \text{ that decides } L \text{ in time } O(n^k)\}$$

(Def.)

$$\mathbf{NP} \;:=\; \bigcup_{k \geqslant 1} \textsc{Ntime}[n^k]$$

(Note) The class **NP** is closed under union and intersection.

The class $\textsc{Ntime}[n^k]$ and **NP**

(Def.) For a fixed integer $k \in \mathbb{N}$:

$$\textsc{Ntime}[n^k] \; := \; \{L \;\mid\; \text{there is an } \underline{\text{NTM}} \; \mathcal{M} \text{ that decides } L \text{ in time } O(n^k)\}$$

(Def.)

$$\mathbf{NP} \; := \; \bigcup_{k \geqslant 1} \textsc{Ntime}[n^k]$$

(Note) The class **NP** is closed under union and intersection.

It is not known whether **NP** is closed under complement.

The class coNP

(Def.)

$$\mathbf{coNP} \; := \; \{L \mid \Sigma^* - L \in \mathbf{NP}\}$$

The class coNP

(Def.)

$$\mathbf{coNP} \;:=\; \{L \;\mid\; \Sigma^* - L \in \mathbf{NP}\}$$

By definition, $\mathbf{NP} = \mathbf{coNP}$ if and only if $\mathbf{NP}$ is closed under complement.

The class coNP

(Def.)

$$\textbf{coNP} := \{L \mid \Sigma^* - L \in \textbf{NP}\}$$

By definition, $\textbf{NP} = \textbf{coNP}$ if and only if **NP** is closed under complement.

(Note) This is NOT the correct definition of **coNP**:

$$L \in \textbf{coNP} \quad \text{if and only if} \quad L \notin \textbf{NP}$$

SAT $\in$ NP

SAT

Input: A propositional formula φ.
Task: Output `True`, if φ has (at least one) satisfying assignment. Otherwise, output `False`.

SAT $\in$ NP

SAT

Input: A propositional formula φ.
Task: Output `True`, if φ has (at least one) satisfying assignment.
Otherwise, output `False`.

(Algo.) On input formula φ:

- Let $x_1, \ldots, x_n$ be the variables in φ.
- For each $i = 1, \ldots, n$ do:
  - $z \ := \ 0 \parallel 1;$
  - If $z == 1$, assign x_i with `True`.
  - If $z == 0$, assign x_i with `False`.
- Check if the formula φ evaluates to true under the assignment.
- If it evaluates to `True`, then ACCEPT.
  If it evaluates to `False`, then REJECT.

$\overline{\textsf{SAT}} \in \textsf{coNP}$

$\overline{\text{SAT}}$

Input: A propositional formula φ.

Task: Output `True`, if φ does not have any satisfying assignment.
Otherwise, output `False`.

$\overline{\text{SAT}} \in$ coNP

$\overline{\text{SAT}}$

Input: A propositional formula φ.
Task: Output `True`, if φ does not have any satisfying assignment.
Otherwise, output `False`.

Since SAT $\in$ **NP**, $\overline{\text{SAT}} \in$ **coNP**.

Some open problems in computer science

Some open problems in computer science

- Is $\mathbf{P} = \mathbf{NP}$?

Some open problems in computer science

- Is $\mathbf{P} = \mathbf{NP}$?
  By definition, $\mathbf{P} \subseteq \mathbf{NP}$.

Some open problems in computer science

- Is $\mathbf{P} = \mathbf{NP}$?
  By definition, $\mathbf{P} \subseteq \mathbf{NP}$.

- Is $\mathbf{NP} = \mathbf{coNP}$?

Some open problems in computer science

- Is $\mathbf{P} = \mathbf{NP}$?
  By definition, $\mathbf{P} \subseteq \mathbf{NP}$.

- Is $\mathbf{NP} = \mathbf{coNP}$?

- Is $\mathbf{P} = \mathbf{NP} \cap \mathbf{coNP}$?

Some open problems in computer science

- Is $\mathbf{P} = \mathbf{NP}$?
  By definition, $\mathbf{P} \subseteq \mathbf{NP}$.
- Is $\mathbf{NP} = \mathbf{coNP}$?
- Is $\mathbf{P} = \mathbf{NP} \cap \mathbf{coNP}$?
  By definition, $\mathbf{P} \subseteq \mathbf{NP} \cap \mathbf{coNP}$.

Table of contents

Polynomial space complexity

(Def.) Let $k \geqslant 1$ be a fixed integer.

A DTM/NTM $\mathcal{M}$ uses space $O(n^k)$, if:

Polynomial space complexity

(Def.) Let $k \geqslant 1$ be a fixed integer.

A DTM/NTM $\mathcal{M}$ uses space $O(n^k)$, if:

- $\mathcal{M}$ halts on every input word $w \in \Sigma^*$.

Polynomial space complexity

(Def.) Let $k \geqslant 1$ be a fixed integer.

A DTM/NTM $\mathcal{M}$ uses space $O(n^k)$, if:

- $\mathcal{M}$ halts on every input word $w \in \Sigma^*$.
- There is $c, n_0 \in \mathbb{N}$ such that for every word $w \in \Sigma^*$ with length $|w| \geqslant n_0$: *each configuration* in every run of $\mathcal{M}$ on w has length $\leqslant c|w|^k$.

Polynomial space complexity

(Def.) Let $k \geqslant 1$ be a fixed integer.

A DTM/NTM $\mathcal{M}$ uses space $O(n^k)$, if:

- $\mathcal{M}$ halts on every input word $w \in \Sigma^*$.
- There is $c, n_0 \in \mathbb{N}$ such that for every word $w \in \Sigma^*$ with length $|w| \geqslant n_0$: *each configuration* in every run of $\mathcal{M}$ on w has length $\leqslant c|w|^k$.

That is, for every run of $\mathcal{M}$ on w with $|w| \geqslant n_0$:

$$C_0 \ \vdash \ C_1 \ \vdash \ \cdots \ \vdash \ C_N$$

C_N can be acc./rej.

the length $|C_i| \leqslant c|w|^k$, for each $i = 0, \ldots, N$.

The class $\textsc{Dspace}[n^k]$ and PSPACE

(Def.) A *DTM/NTM $\mathcal{M}$ decides/accepts a language L in space $O(n^k)$*, if:

- $\mathcal{M}$ decides L.
- $\mathcal{M}$ uses space $O(n^k)$.

The class $\text{DSPACE}[n^k]$ and PSPACE

(Def.) A *DTM/NTM $\mathcal{M}$ decides/accepts a language L in space $O(n^k)$*, if:

- $\mathcal{M}$ decides L.
- $\mathcal{M}$ uses space $O(n^k)$.

(Def.) For a fixed integer $k \in \mathbb{N}$:

$$\text{DSPACE}[n^k] := \{L \mid \text{there is a } \underline{\text{DTM}}\ \mathcal{M} \text{ that decides } L \text{ in space } O(n^k)\}$$

The class $\textsc{Dspace}[n^k]$ and PSPACE

(Def.) A *DTM/NTM $\mathcal{M}$ decides/accepts a language L in space $O(n^k)$*, if:

- $\mathcal{M}$ decides L.
- $\mathcal{M}$ uses space $O(n^k)$.

(Def.) For a fixed integer $k \in \mathbb{N}$:

$$\textsc{Dspace}[n^k] \;:=\; \{L \mid \text{there is a } \underline{\text{DTM}}\ \mathcal{M} \text{ that decides } L \text{ in space } O(n^k)\}$$

(Def.)

$$\textbf{PSPACE} \;:=\; \bigcup_{k \geqslant 1} \textsc{Dspace}[n^k]$$

The class $\textsc{Dspace}[n^k]$ and PSPACE

(Def.) A *DTM/NTM $\mathcal{M}$ decides/accepts a language L in space $O(n^k)$*, if:

- $\mathcal{M}$ decides L.
- $\mathcal{M}$ uses space $O(n^k)$.

(Def.) For a fixed integer $k \in \mathbb{N}$:

$$\textsc{Dspace}[n^k] \;:=\; \{L \mid \text{there is a } \underline{\text{DTM}}\ \mathcal{M} \text{ that decides } L \text{ in space } O(n^k)\}$$

(Def.)

$$\textbf{PSPACE} \;:=\; \bigcup_{k \geqslant 1} \textsc{Dspace}[n^k]$$

(Note) The class **PSPACE** is closed under complement, union and intersection.

The class $\textsc{Nspace}[n^k]$ and NPSPACE

(Def.) For a fixed integer $k \in \mathbb{N}$:

$$\textsc{Nspace}[n^k] \;:=\; \{L \;\mid\; \text{there is an } \underline{\text{NTM}}\ \mathcal{M} \text{ that decides } L \text{ in space } O(n^k)\}$$

The class $\textsc{Nspace}[n^k]$ and NPSPACE

(Def.) For a fixed integer $k \in \mathbb{N}$:

$$\textsc{Nspace}[n^k] \ := \ \{L \mid \text{there is an } \underline{\text{NTM}}\ \mathcal{M} \text{ that decides } L \text{ in space } O(n^k)\}$$

(Def.)

$$\textbf{NPSPACE} \ := \ \bigcup_{k \geqslant 1} \textsc{Nspace}[n^k]$$

The class $\textsc{Nspace}[n^k]$ and NPSPACE

(Def.) For a fixed integer $k \in \mathbb{N}$:

$$\textsc{Nspace}[n^k] \;:=\; \{L \mid \text{there is an } \underline{\text{NTM}}\ \mathcal{M} \text{ that decides } L \text{ in space } O(n^k)\}$$

(Def.)

$$\textbf{NPSPACE} \;:=\; \bigcup_{k \geqslant 1} \textsc{Nspace}[n^k]$$

(Def.)

$$\textbf{coNPSPACE} \;:=\; \{L \mid \Sigma^* - L \in \textbf{NPSPACE}\}$$

What is known and not known so far

It is known that:

What is known and not known so far

It is known that:

- If $L \in \textsc{Nspace}[n^k]$, then $\Sigma^* - L \in \textsc{Nspace}[n^k]$.

What is known and not known so far

It is known that:

- If $L \in \text{NSPACE}[n^k]$, then $\Sigma^* - L \in \text{NSPACE}[n^k]$.
- $\text{NSPACE}[n^k] \subseteq \text{DSPACE}[n^{2k}]$.

Thus,

$$\textbf{PSPACE} = \textbf{NPSPACE} = \textbf{coNPSPACE}$$

What is known and not known so far

It is known that:

- If $L \in \text{NSPACE}[n^k]$, then $\Sigma^* - L \in \text{NSPACE}[n^k]$.
- $\text{NSPACE}[n^k] \subseteq \text{DSPACE}[n^{2k}]$.

Thus,

$$\textbf{PSPACE} = \textbf{NPSPACE} = \textbf{coNPSPACE}$$

In literature we usually only use **PSPACE**.

The notations **NPSPACE** and **coNPSPACE** are hardly used.

What is known and not known so far

It is known that:

- If $L \in \text{NSPACE}[n^k]$, then $\Sigma^* - L \in \text{NSPACE}[n^k]$.
- $\text{NSPACE}[n^k] \subseteq \text{DSPACE}[n^{2k}]$.

Thus,

$$\textbf{PSPACE} = \textbf{NPSPACE} = \textbf{coNPSPACE}$$

In literature we usually only use **PSPACE**.

The notations **NPSPACE** and **coNPSPACE** are hardly used.

By definition:

$$\textbf{P} \subseteq \textbf{NP} \subseteq \textbf{PSPACE}$$

What is known and not known so far

It is known that:

- If $L \in \textsc{Nspace}[n^k]$, then $\Sigma^* - L \in \textsc{Nspace}[n^k]$.
- $\textsc{Nspace}[n^k] \subseteq \textsc{Dspace}[n^{2k}]$.

Thus,

$$\textbf{PSPACE} = \textbf{NPSPACE} = \textbf{coNPSPACE}$$

In literature we usually only use **PSPACE**.

The notations **NPSPACE** and **coNPSPACE** are hardly used.

By definition:

$$\textbf{P} \subseteq \textbf{NP} \subseteq \textbf{PSPACE}$$

We do not know whether any of the inclusion is strict.

Table of contents

The class **L**

(Def.) A language L is in **L**, if there is a 2-tape DTM $\mathcal{M}$ that decides L and there is $c \in \mathbb{N}$ such that for every input word w:

- The first tape always contains only the input word w.
  That is, $\mathcal{M}$ can only read the first tape, but never changes the content of the first tape.
- $\mathcal{M}$ uses space $\leqslant c \cdot \log(|w|)$ in its second tape.

The class L

(Def.) A language L is in **L**, if there is a 2-tape DTM $\mathcal{M}$ that decides L and there is $c \in \mathbb{N}$ such that for every input word w:

- The first tape always contains only the input word w.
  That is, $\mathcal{M}$ can only read the first tape, but never changes the content of the first tape.
- $\mathcal{M}$ uses space $\leqslant c \cdot \log(|w|)$ in its second tape.

(Note) The 2-tape DTM requirement is not strict. It can be replaced with multiple tape DTM with the condition that the TM does not change the content of the first tape and the number of cells used in the other tapes is $\leqslant c \cdot \log(|w|)$.

The class NL and coNL

(Def.) The class **NL** is defined similarly as **L**, with the difference that the Turing machine $\mathcal{M}$ is a 2-tape NTM.

The class NL and coNL

(Def.) The class **NL** is defined similarly as **L**, with the difference that the Turing machine $\mathcal{M}$ is a 2-tape NTM.

(Note) Again, the 2-tape requirement is not strict and can be replaced with multiple tape NTM.

The class NL and coNL

(Def.) The class **NL** is defined similarly as **L**, with the difference that the Turing machine $\mathcal{M}$ is a 2-tape NTM.

(Note) Again, the 2-tape requirement is not strict and can be replaced with multiple tape NTM.

(Def.)

$$\textbf{coNL} := \{L \mid \Sigma^* - L \in \textbf{NL}\}$$

What is known and not known so far

(Deterministic/non-deterministic time/space hierarchy theorem) For every $k \geqslant 1$ and $\epsilon > 0$:

$$\mathrm{DTIME}[n^k] \subsetneq \mathrm{DTIME}[n^{k+\epsilon}] \qquad \mathrm{NTIME}[n^k] \subsetneq \mathrm{NTIME}[n^{k+\epsilon}]$$

$$\mathrm{DSPACE}[n^k] \subsetneq \mathrm{DSPACE}[n^{k+\epsilon}] \qquad \mathrm{NSPACE}[n^k] \subsetneq \mathrm{NSPACE}[n^{k+\epsilon}]$$

It is known that:

What is known and not known so far

(Deterministic/non-deterministic time/space hierarchy theorem) For every $k \geqslant 1$ and $\epsilon > 0$:

$$\textsc{Dtime}[n^k] \subsetneq \textsc{Dtime}[n^{k+\epsilon}] \qquad \textsc{Ntime}[n^k] \subsetneq \textsc{Ntime}[n^{k+\epsilon}]$$

$$\textsc{Dspace}[n^k] \subsetneq \textsc{Dspace}[n^{k+\epsilon}] \qquad \textsc{Nspace}[n^k] \subsetneq \textsc{Nspace}[n^{k+\epsilon}]$$

It is known that:

- **NL** $\subseteq$ **P**.

What is known and not known so far

(Deterministic/non-deterministic time/space hierarchy theorem) For every $k \geqslant 1$ and $\epsilon > 0$:

$$\textsc{Dtime}[n^k] \subsetneq \textsc{Dtime}[n^{k+\epsilon}] \qquad \textsc{Ntime}[n^k] \subsetneq \textsc{Ntime}[n^{k+\epsilon}]$$

$$\textsc{Dspace}[n^k] \subsetneq \textsc{Dspace}[n^{k+\epsilon}] \qquad \textsc{Nspace}[n^k] \subsetneq \textsc{Nspace}[n^{k+\epsilon}]$$

It is known that:

- **NL** $\subseteq$ **P**.
- **NL** $=$ **coNL**.

What is known and not known so far

(Deterministic/non-deterministic time/space hierarchy theorem) For every $k \geqslant 1$ and $\epsilon > 0$:

$$\textsc{Dtime}[n^k] \subsetneq \textsc{Dtime}[n^{k+\epsilon}] \qquad \textsc{Ntime}[n^k] \subsetneq \textsc{Ntime}[n^{k+\epsilon}]$$

$$\textsc{Dspace}[n^k] \subsetneq \textsc{Dspace}[n^{k+\epsilon}] \qquad \textsc{Nspace}[n^k] \subsetneq \textsc{Nspace}[n^{k+\epsilon}]$$

It is known that:

- **NL** $\subseteq$ **P**.
- **NL** $=$ **coNL**.
- **L** $\subsetneq$ **PSPACE**

What is known and not known so far

(Deterministic/non-deterministic time/space hierarchy theorem) For every $k \geqslant 1$ and $\epsilon > 0$:

$$\textsc{Dtime}[n^k] \subsetneq \textsc{Dtime}[n^{k+\epsilon}] \qquad \textsc{Ntime}[n^k] \subsetneq \textsc{Ntime}[n^{k+\epsilon}]$$

$$\textsc{Dspace}[n^k] \subsetneq \textsc{Dspace}[n^{k+\epsilon}] \qquad \textsc{Nspace}[n^k] \subsetneq \textsc{Nspace}[n^{k+\epsilon}]$$

It is known that:

- $\mathbf{NL} \subseteq \mathbf{P}$.
- $\mathbf{NL} = \mathbf{coNL}$.
- $\mathbf{L} \subsetneq \mathbf{PSPACE}$

  Because $\mathbf{L} \subseteq \textsc{Dspace}[n] \subsetneq \textsc{Dspace}[n^2] \subseteq \mathbf{PSPACE}$.

What is known and not known so far

(Deterministic/non-deterministic time/space hierarchy theorem) For every $k \geqslant 1$ and $\epsilon > 0$:

$$\text{DTIME}[n^k] \subsetneq \text{DTIME}[n^{k+\epsilon}] \qquad \text{NTIME}[n^k] \subsetneq \text{NTIME}[n^{k+\epsilon}]$$

$$\text{DSPACE}[n^k] \subsetneq \text{DSPACE}[n^{k+\epsilon}] \qquad \text{NSPACE}[n^k] \subsetneq \text{NSPACE}[n^{k+\epsilon}]$$

It is known that:

- **NL** $\subseteq$ **P**.
- **NL** $=$ **coNL**.
- **L** $\subsetneq$ **PSPACE**

  Because **L** $\subseteq \text{DSPACE}[n] \subsetneq \text{DSPACE}[n^2] \subseteq$ **PSPACE**.
- Likewise, **NL** $\subsetneq$ **PSPACE**.

What is known and not known so far

(Deterministic/non-deterministic time/space hierarchy theorem) For every $k \geqslant 1$ and $\epsilon > 0$:

$$\mathrm{DTIME}[n^k] \subsetneq \mathrm{DTIME}[n^{k+\epsilon}] \qquad \mathrm{NTIME}[n^k] \subsetneq \mathrm{NTIME}[n^{k+\epsilon}]$$

$$\mathrm{DSPACE}[n^k] \subsetneq \mathrm{DSPACE}[n^{k+\epsilon}] \qquad \mathrm{NSPACE}[n^k] \subsetneq \mathrm{NSPACE}[n^{k+\epsilon}]$$

It is known that:

- **NL** $\subseteq$ **P**.
- **NL** $=$ **coNL**.
- **L** $\subsetneq$ **PSPACE**

  Because **L** $\subseteq \mathrm{DSPACE}[n] \subsetneq \mathrm{DSPACE}[n^2] \subseteq$ **PSPACE**.
- Likewise, **NL** $\subsetneq$ **PSPACE**.

  Likewise, **NL** $\subseteq \mathrm{NSPACE}[n] \subseteq \mathrm{DSPACE}[n^2] \subsetneq \mathrm{DSPACE}[n^3] \subseteq$ **PSPACE**.

What is known and not known so far – continued

Putting all the pieces together:

$$\mathbf{L} \subseteq \mathbf{NL} \subseteq \mathbf{P} \subseteq \mathbf{NP} \subseteq \mathbf{PSPACE}$$

Since $\mathbf{L} \subsetneq \mathbf{PSPACE}$ and $\mathbf{NL} \subsetneq \mathbf{PSPACE}$, at least one of the inclusions is strict.

What is known and not known so far – continued

Putting all the pieces together:

$$\mathbf{L} \subseteq \mathbf{NL} \subseteq \mathbf{P} \subseteq \mathbf{NP} \subseteq \mathbf{PSPACE}$$

Since $\mathbf{L} \subsetneq \mathbf{PSPACE}$ and $\mathbf{NL} \subsetneq \mathbf{PSPACE}$, at least one of the inclusions is strict.

The question: Which one?

End of Lesson 10