Large-scale Knowledge Processing
Lecture 2

Kazuhisa Seto

Today's Lecture

Study the foundation on theory of computation

- Turing Machine
- Time Complexity
- Class P and Class NP

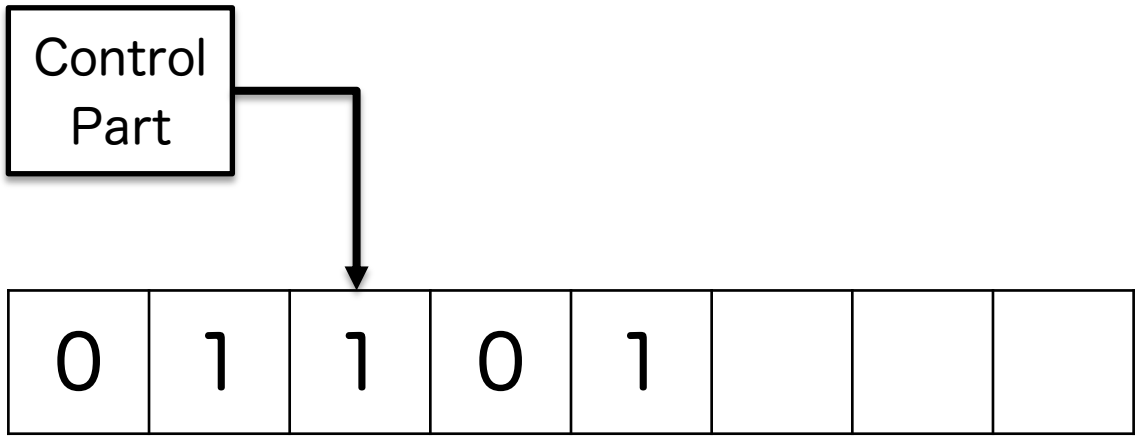
It is important to know the definitions of NP-hard and NP-complete that are used for many areas.

Turing Machine

Turing Machine

One of mathematical model for computers

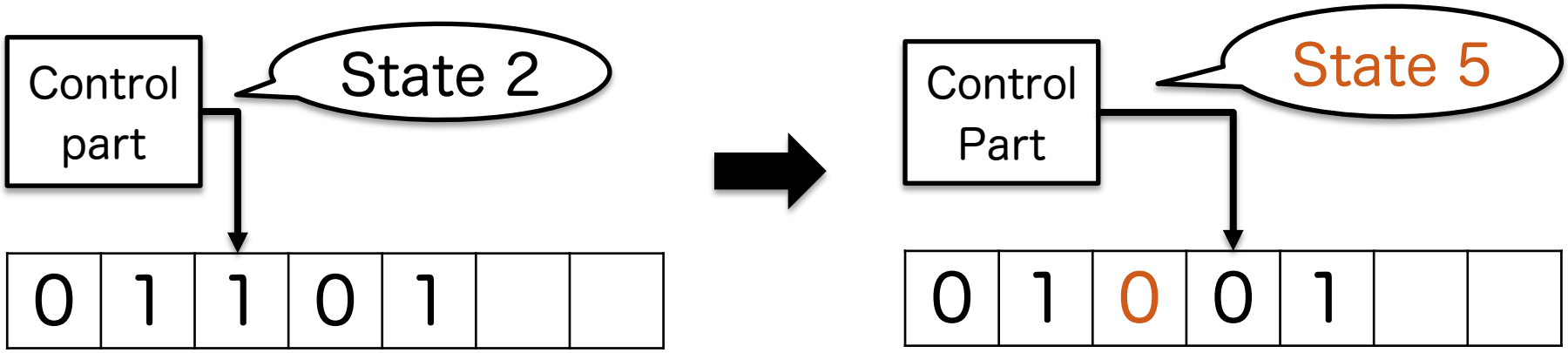
- Turing Machine (TM) have control part and writable tape.
- The tape has an infinite length.



Turing Machine (TM)

Computation by Turing Machine

- Control part read/write character on the current cell of the tape.
- According to a character and the state in control part, control part changes to its state and move on the tape.



An Example of computation by TM

Consider a language $L = \{ w\#w \mid w \in \{0,1\}^* \}$

If a string s on tape belongs to L, TM accepts s.

Otherwise, TM reject s.

A) 1010#1010 → accept

B) 01010#10101 → reject

C) 0101#10101 → reject

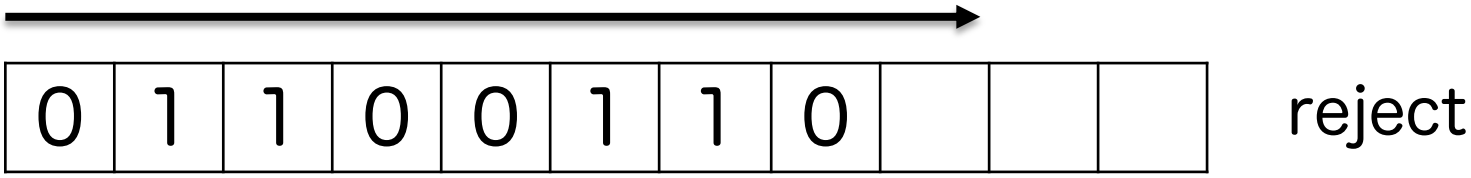
D) 11111#1111 → reject

E) 00000000 → reject

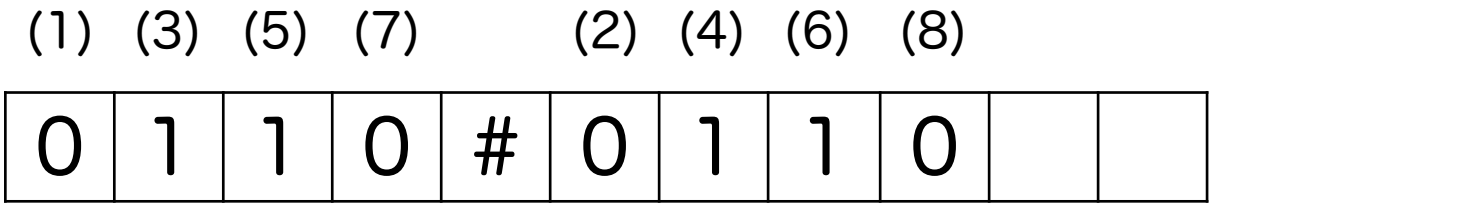
An Example of computation by TM

By Turing Machine, how to determine whether a string s belongs to the language $L = \{ w\#w \mid w \in \{0,1\}^* \}$ or not ?

- First, TM moves head of control part from left to right.
  - If there exists no #, TM reject s.



- Otherwise, head of control part move to the leftmost, check a character in the following order.

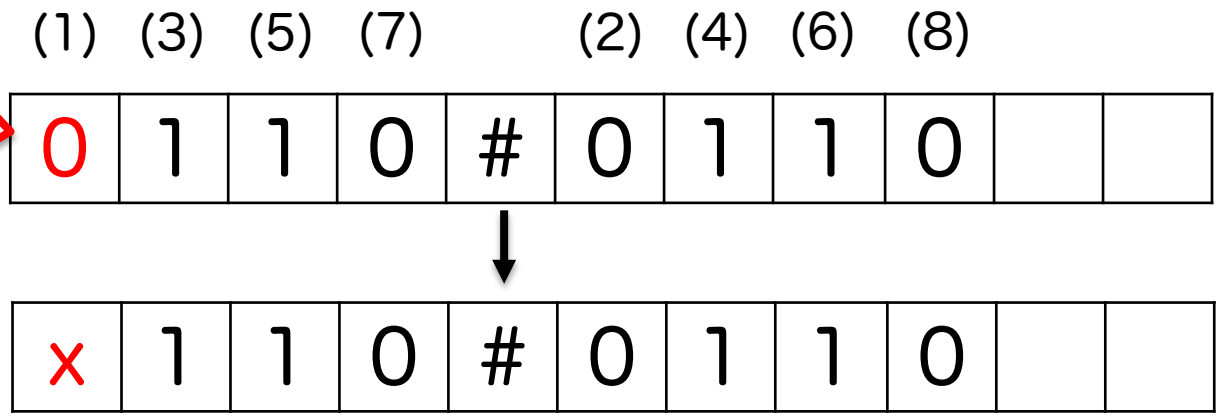


An Example of computation by TM

TM checks (1)'s symbol.

- memorize the symbol and replace it by the symbol X.

Memorize the symbol 0 and replace it by the symbol x.

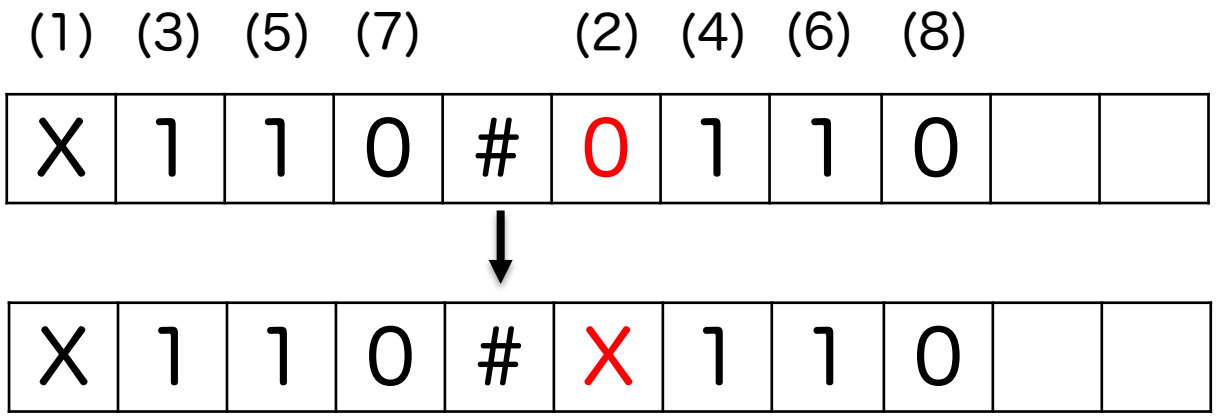


Next, TM moves to (2) and checks whether the (2)'s symbol is same as (1)'s symbol.

An Example of computation by TM

If the (2)'s symbol is not same the (1)'s symbol, TM outputs Reject.

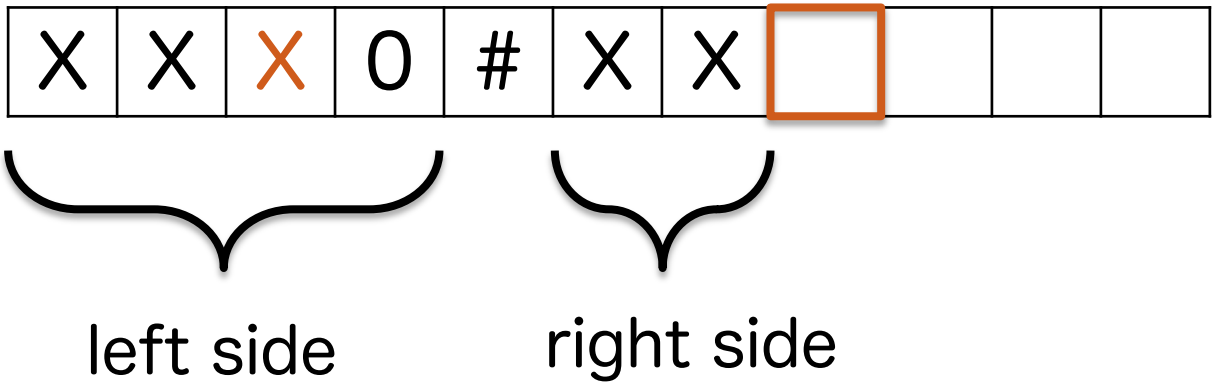
Otherwise, TM replace the (2)'s symbol by the symbol X and move to (3).



TM repeats these operations.

An Example of computation by TM

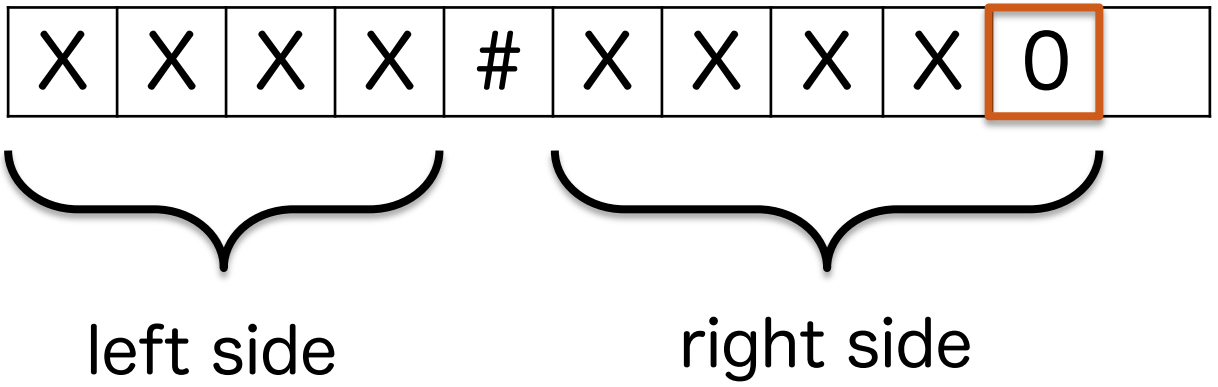
When TM moves to right after checking a symbol on the left side, there is no symbol in the tape, then TM outputs Reject because the number of the symbol on the left side is larger than that on the right side.



An Example of computation by TM

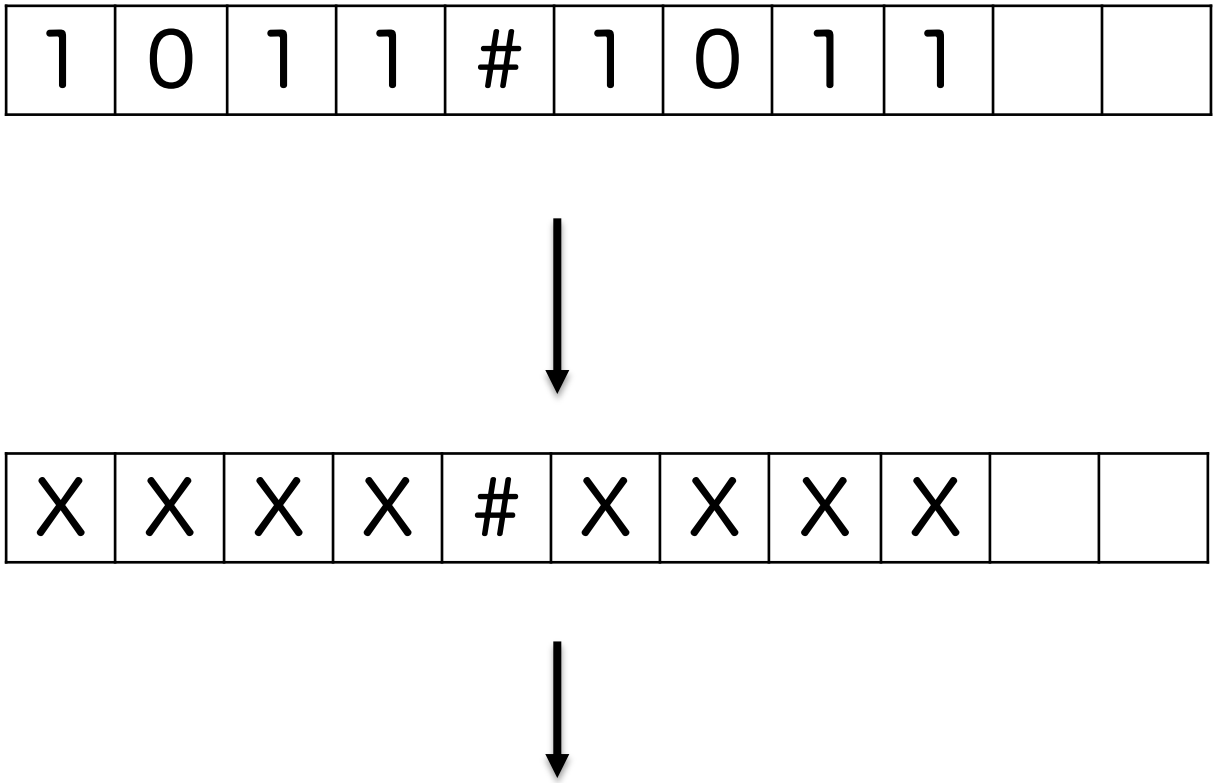
When TM have checked all symbols on the left side, TM moves to the right side and checks whether no symbol remains.

If it remains, TM outputs Reject.



An Example of computation by TM

- Otherwise, TM outputs Accept.



Time Complexity

The number of steps

How many steps TM accepts or rejects a language?

- TM takes one step for moving on one cell.
- TM takes one step for accessing a symbol on the tape.
- TM takes one step for memorizing a symbol.

etc...

We need to count all steps one by one.

It is quite hard to count correctly.

Time Complexity

The number of steps that TM needs for recognizing a language L

- The number of steps is represented by the input length.
  - The length of string initially written on tapes
- We consider time complexity when the worst case or the average case. In this lecture, we consider the worst case.
  - ✓ Worst Case: The largest number of steps between all inputs.

Landau Notation

It is hard to count exact number of steps of TM .

→ We count it a little roughly.

Examples

- Adding two 1-bit numbers is possible in a constant steps.
  - We represent its number of steps by $O(1)$ steps.
- For checking all the symbol on the tape of length n, we need at least n steps.
  - We represent its number of steps by by $\Omega(n)$ steps.

The definition of Landau Notation
Big-O notation

$f(n) = O(g(n))$ (or $f(n) \in O(g(n))$)

if there exists a natural number n_0 and a positive number c such that for all natural numbers $n \geq n_0$, $f(n) \leq cg(n)$ holds.

【Example】

We consider $f(n) = 3n^2 + 2n + 2, g(n) = n^2$.

Setting $c = 4, n_0 = 3$, $f(n_0) = 35$、$cg(n_0) = 36$.

Then, for $n \geq 3$, $f(n) \leq cg(n)$ holds.

Thus, $3n^2 + 2n + 2 = O(n^2)$.

The definition of Landau Notation
Big-Omega notation

$f(n) = \Omega(g(n))$ (or $f(n) \in \Omega(g(n))$)

if there exists a natural number n_0 and a positive number c such that for all natural numbers $n \geq n_0$, $f(n) \geq cg(n)$ holds.

【Example】

We consider $f(n) = n^2 + n - 2, g(n) = n^2$.

Setting $c = 1, n_0 = 2,\ f(n_0) = 4$、$cg(n_0) = 4$.

Then, for $n \geq 2$, $f(n) \geq cg(n)$ holds.

Thus, $n^2 + n - 2 = \Omega(n^2)$.

The definition of Landau Notation

Intuitive meanings of O(·), Ω(·)

- $f(n) = O\big(g(n)\big)$

  $f(n)$ increases at most as fast as $g(n)$

- $f(n) = \Omega\big(g(n)\big)$

  $f(n)$ increases at least as fast as $g(n)$

Both $f(n) = O\big(g(n)\big)$ and $f(n) = \Omega\big(g(n)\big)$ holds,

then $f(n) = \Theta\big(g(n)\big)$.

How to compute big-O and big-Omega Representation.

- Pick up the fastest increasing function and remove the other functions.
- Replace coefficient by 1.

【例】

- $f(n) = 3n^4 + 2n^2 + 3 = O(n^4)$
- $f(n) = 0.2n\log_2 n + 4n - 2 = O(n\log_2 n)$
- $f(n) = 3 \cdot 2^{0.2n} + 2n^{\log_2 n} = 3 \cdot 2^{0.2n} + 2^{(\log_2 n)^2+1} = O(2^{0.2n})$

Exercise 1

Represent each function $f(n)$ by a Big-O representation.

- $f(n) = 3$
- $f(n) = 2\log_2 n + n + 1$
- $f(n) = 3\sqrt{n} + n^{0.3}\log_2 n$
- $f(n) = \log_2 n + n^{\epsilon}\ (\epsilon > 0)$
- $f(n) = 2^{0.1\sqrt{n}} + 2^n$
- $f(n) = 2^n + n!$

Exercise 1 (Answer)

Represent each function $f(n)$ by a Big-O representation.

- $f(n) = 3 = O(1)$
- $f(n) = 2\log_2 n + n + 1 = O(n)$
- $f(n) = 3\sqrt{n} + n^{0.3}\log_2 n = O(\sqrt{n})$
- $f(n) = \log_2 n + n^{\epsilon}\ (\epsilon > 0) = O(n^{\epsilon})$
- $f(n) = 2^{0.1\sqrt{n}} + 2^n = O(2^n)$
- $f(n) = 2^n + n! = O(n!)$

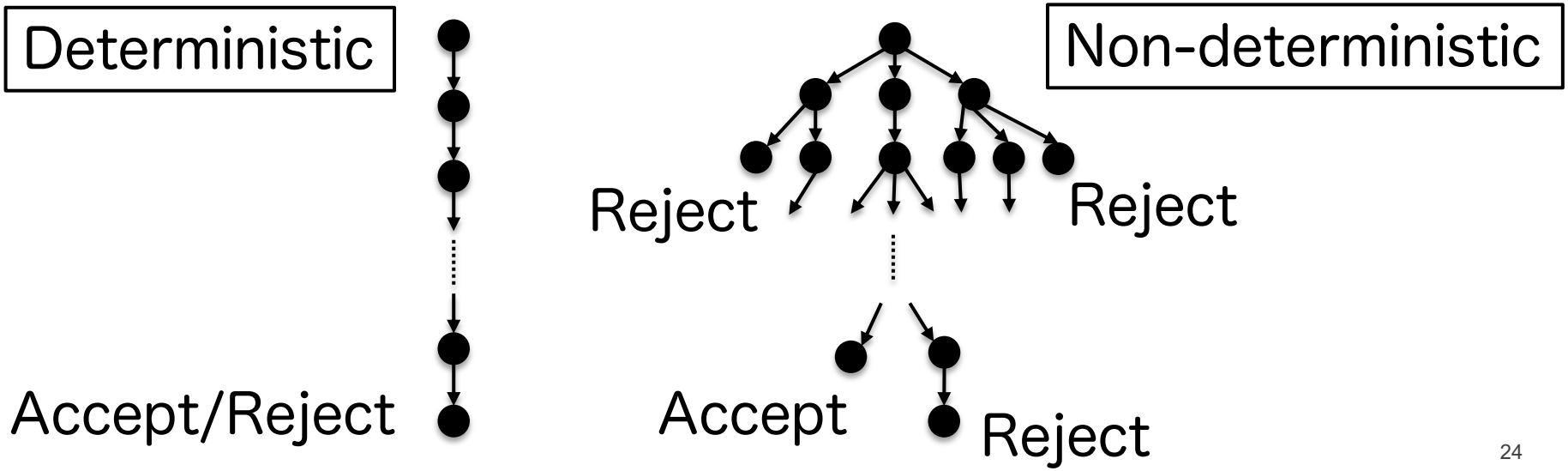
Class P and Class NP

Deterministic and Non-deterministic Computation

The computation by TM has deterministic and non-deterministic.

In computations of TM for a given input

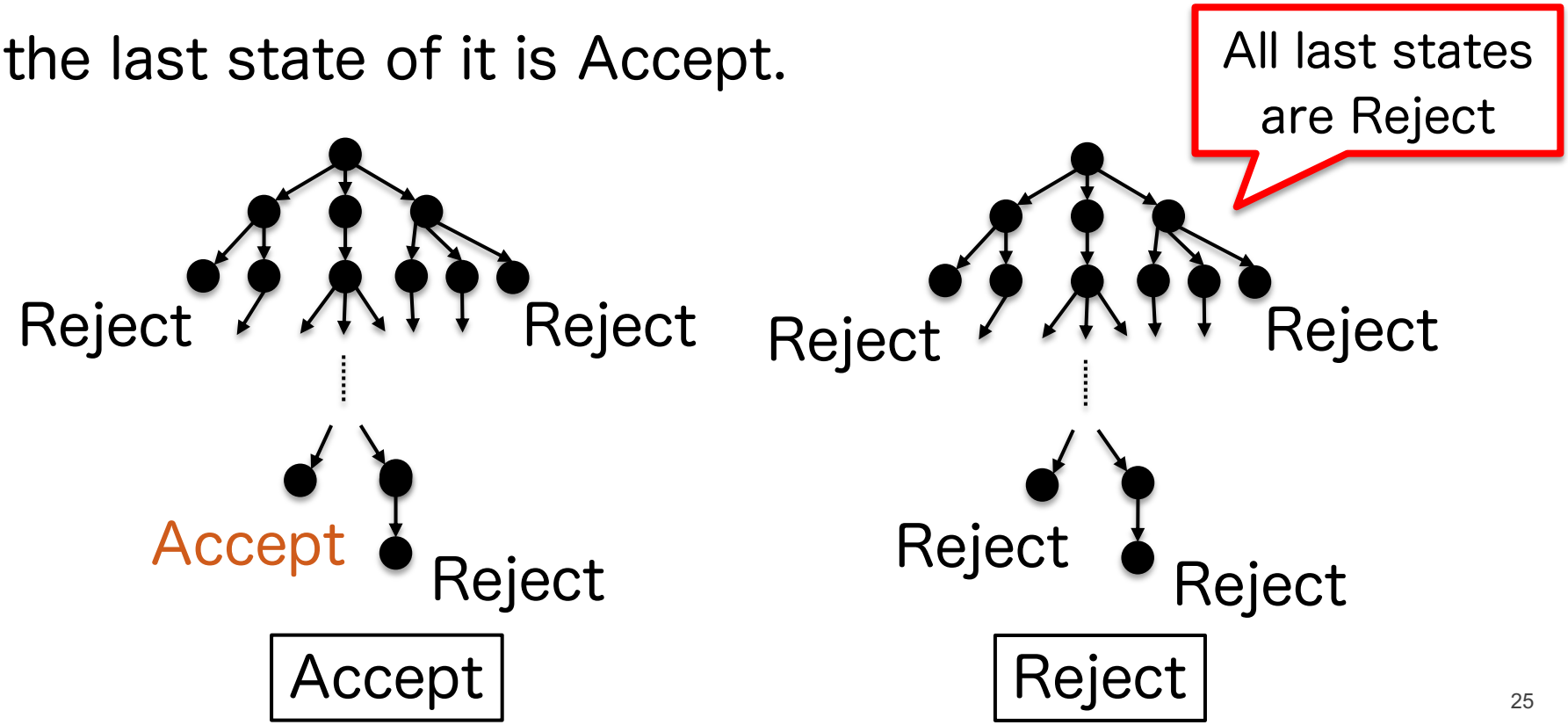
- Deterministic : There is no branch.
- Non-deterministic : There may exists some branches.



The condition of Acceptance by TM

What conditions of acceptance by TM ?

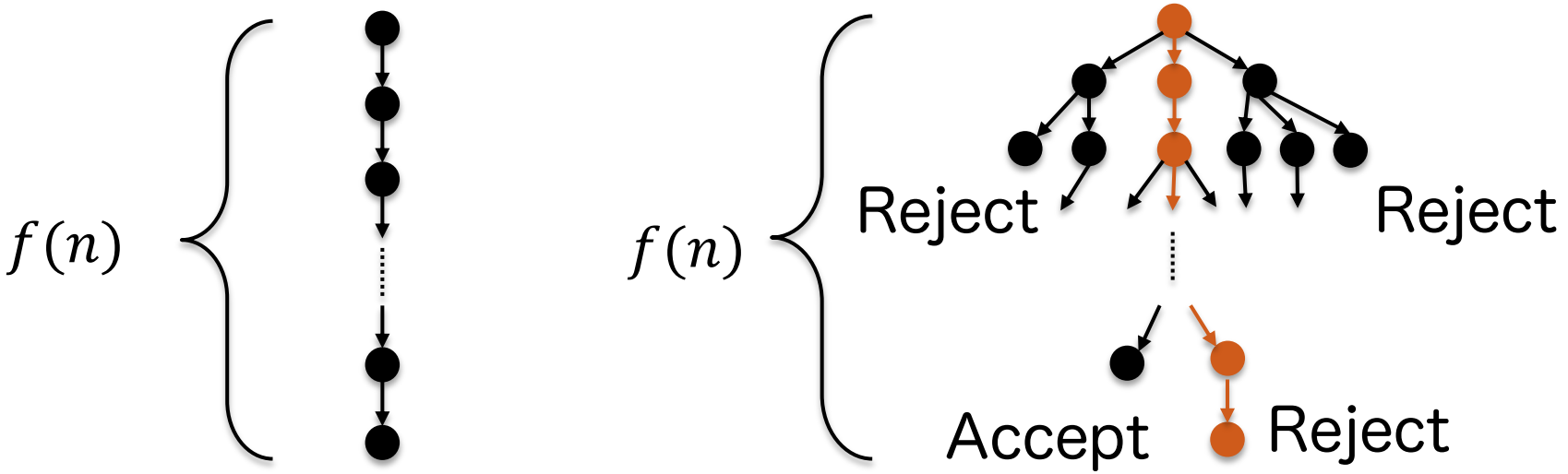
- Deterministic：the last state is Accept.
- Non-deterministic ：There exists some path such that the last state of it is Accept.



Time Complexity

Time Complexity：$f(n)$

- Deterministic：The length of computation path.
- Non-deterministic：The length of the longest computation path between all computation paths.



The definition of decision problem

Decision Problem :

A problem whose answer Yes or No (1 or 0)

decision problem

x	y	z	f
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

x	y	f
1	1	0
1	2	1
1	3	1
2	1	1
2	2	0
2	3	1
3	1	1
3	2	1
3	3	0

Not decision problem

x	y	z	f
0	0	0	1
0	0	1	0
0	1	0	2
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	2
1	1	1	1

Class P and Class NP

Class P（Polynomial time）

- A set of decision problems that can be computed by deterministic TM in polynomial time.

Class NP（Nondeterministic Polynomial time）

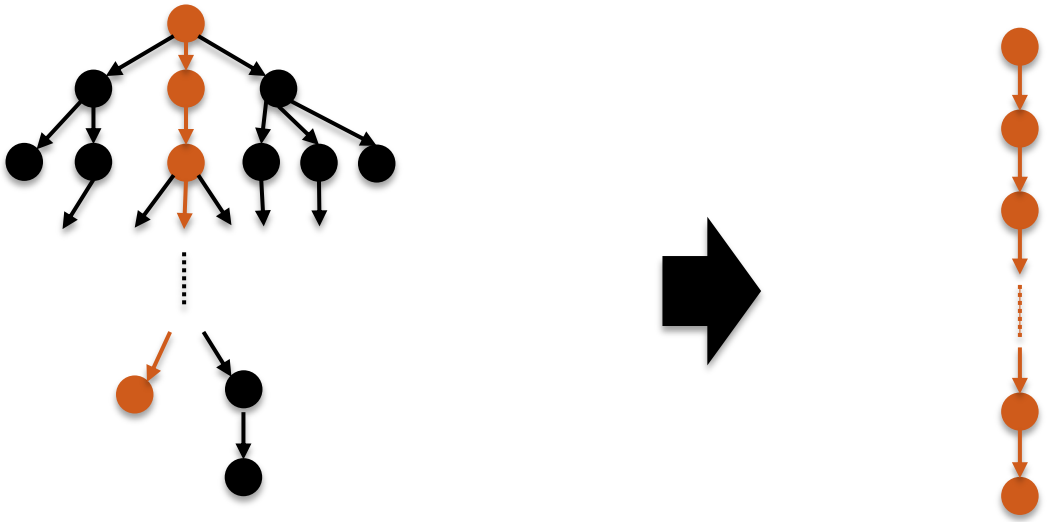
- A set of decision problems that can be computed by non-deterministic TM in polynomial time.

＊ Polynomial time means that time complexity is bounded by some polynomial function such as $O(n^k)$.

Another definition of class NP

Given an information of sequences of computation, non-deterministic computation becomes a deterministic computation.

- For example, when we are given the way of branches, we determine the computation path (orange path)



The formal another definition of NP

A set of decision problems such that

given an input x and witness w, we can check that

x output yes by using w, in $O((|x| + |w|)^k)$ time, where

$|w| = O(|x|^c)$ and c, k are constant.

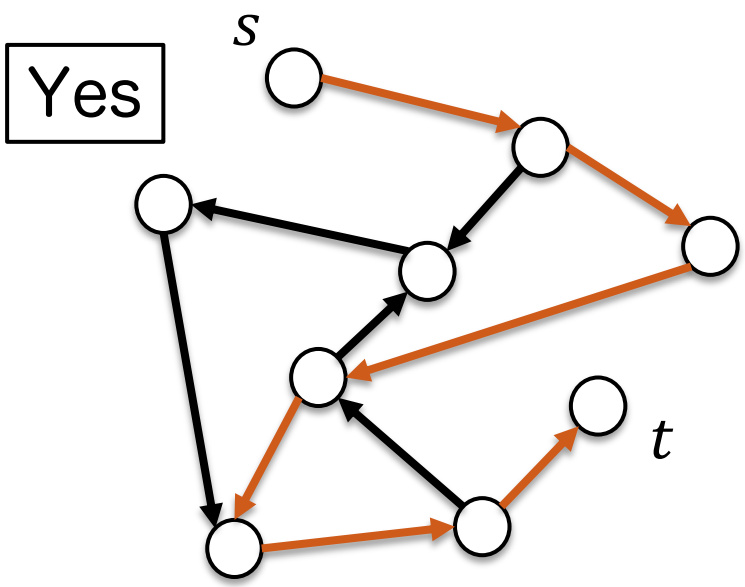
Roughly speaking, a set of decision problems such that when we given an input x of a problem and its solution w, we can check that w is truly correct answer of x in polynomial time.

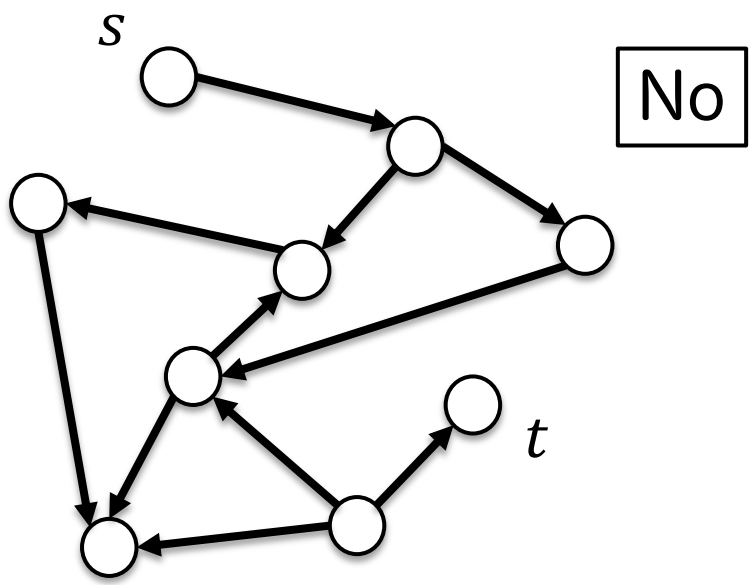
An example of class P

s-t path problem

Input : a directed graph G , vertices s, t.

Ask : Is there a path from s to t ?





By depth-first search, this can be solved in $O(|G|)$ time.

An example of class P

Prime Number Problem

Input : A natural number N

Ask : Is N prime ?

By a simple algorithm, it can be solved in $O(\sqrt{N})$ time, however this algorithm is not polynomial time algorithm.

- When a natural number N is encoded to a binary digit, we need 0/1 $[\log_2 N]$ bit. Thus, input length is $O(\log_2 n)$, then polynomial time means $O\left((\log_2 n)^k\right)$ time.

An example of class P

Prime Number Problem

Input : A natural number N

Ask : Is N prime ?

But, there exists an $O\left((\log_2 n)^{\frac{15}{2}}\right)$ algorithm [Agrawal et al. 2004]. This algorithm is polynomial time algorithm.

An example of class P

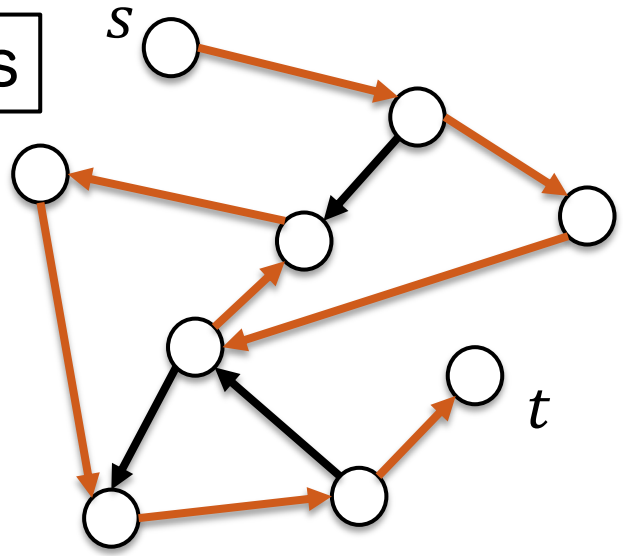
Hamiltonian Path Problem

Input : A directed graph G and vertices s, t.

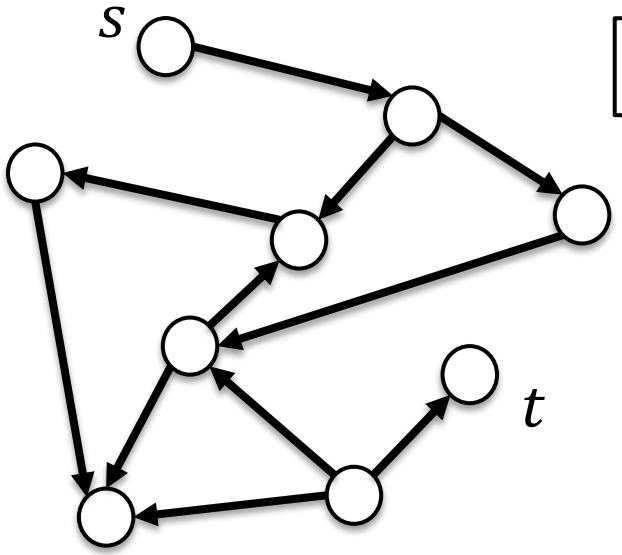
Ask : Is there a Hamiltonian path from s to t ?

- Hamiltonian path : A path that visits every vertex in the graph exactly once.

Yes



No



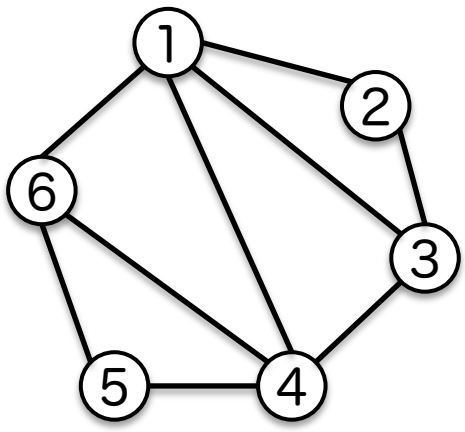
An example of class NP

Graph Isomorphism Problem

Input：two graph G and H

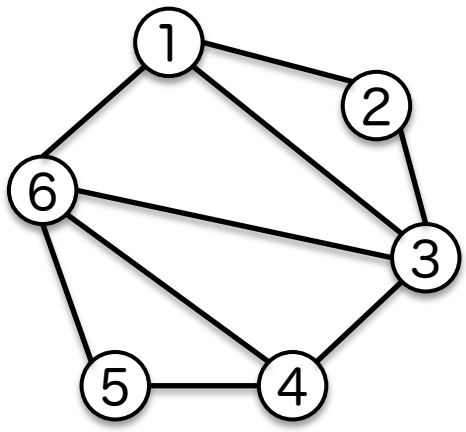
Ask：Are G and H isomorphic？ (By relabeling the vertices of H, G and H are the same drawing.)

1 ⇔ 6
2 ⇔ 5
3 ⇔ 4

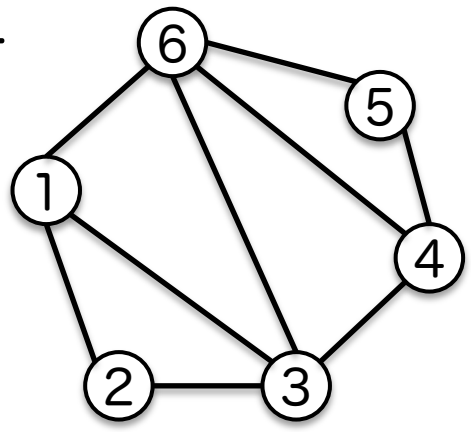


G

Yes



H



P vs. NP Problem

Are Class P and class NP equivalent ?

- It is clear that any problem in P is also in NP.
  - Non-deterministic computations include deterministic computations
- However, it has yet to be shown that some problems exist in NP that do not belong to P.

This is quite challenging problem with a prize of $1 million awarded by the Clay Mathematics Institute.

Summary

A Turing machine is a mathematical model of computer.

- There exist deterministic TM and non-deterministic TM.

Time Complexity

- Landau Notation

Class P and Class NP, P vs. NP

- Examples of Problems in P or NP
- P vs. NP is a big open problem