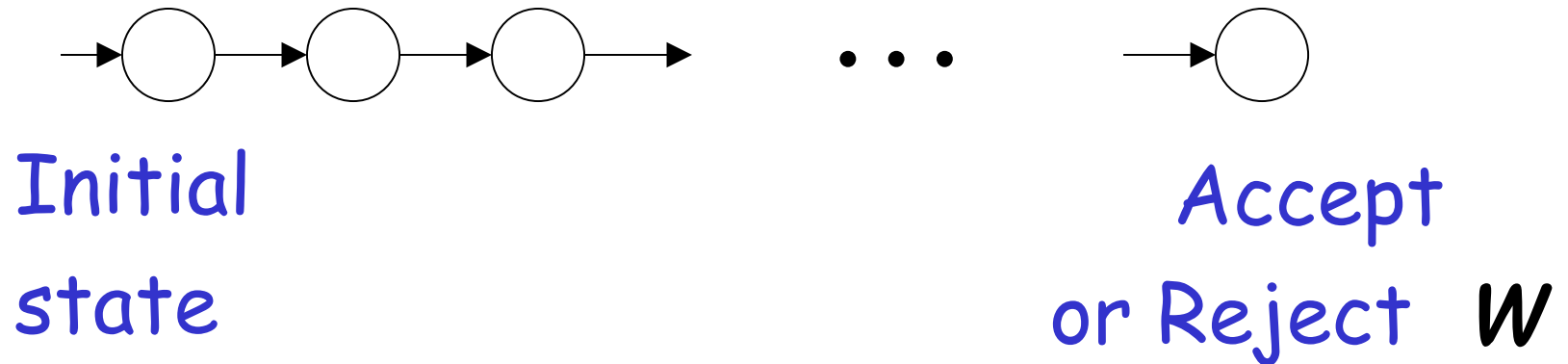


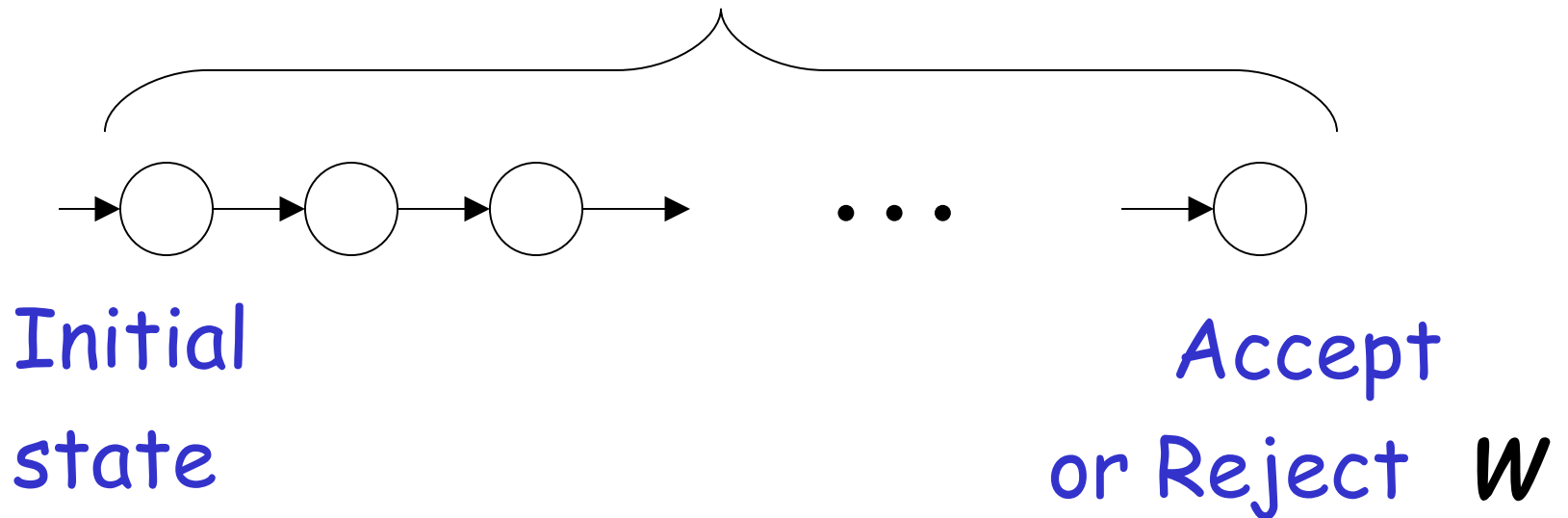
# Time Complexity

Consider a deterministic Turing Machine  $M$   
which decides a language  $L$

For any string  $w$  the computation of  $M$  terminates in a finite amount of transitions

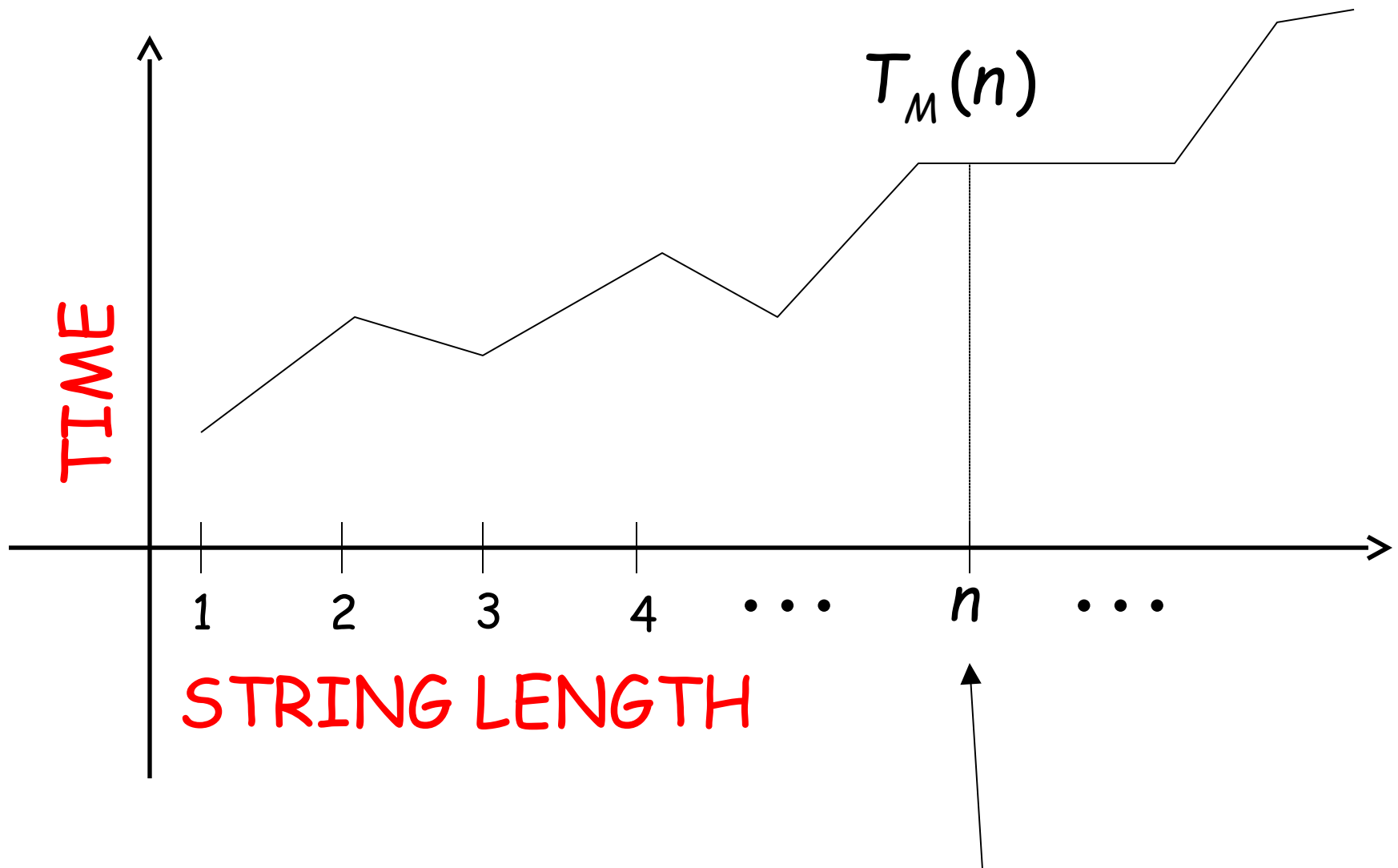


Decision Time = #transitions



Consider now all strings of length  $n$

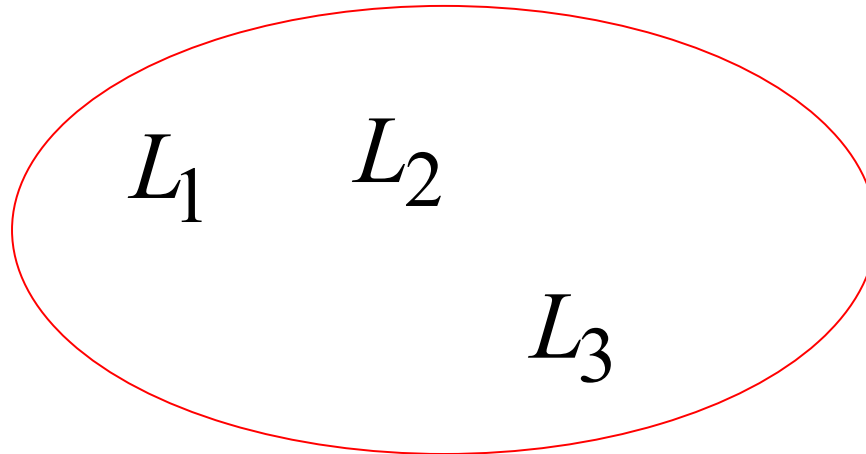
$T_M(n)$  = maximum time required to decide  
any string of length  $n$



Max time to decide string of length  $n$

Time Complexity Class:  $TIME(T(n))$

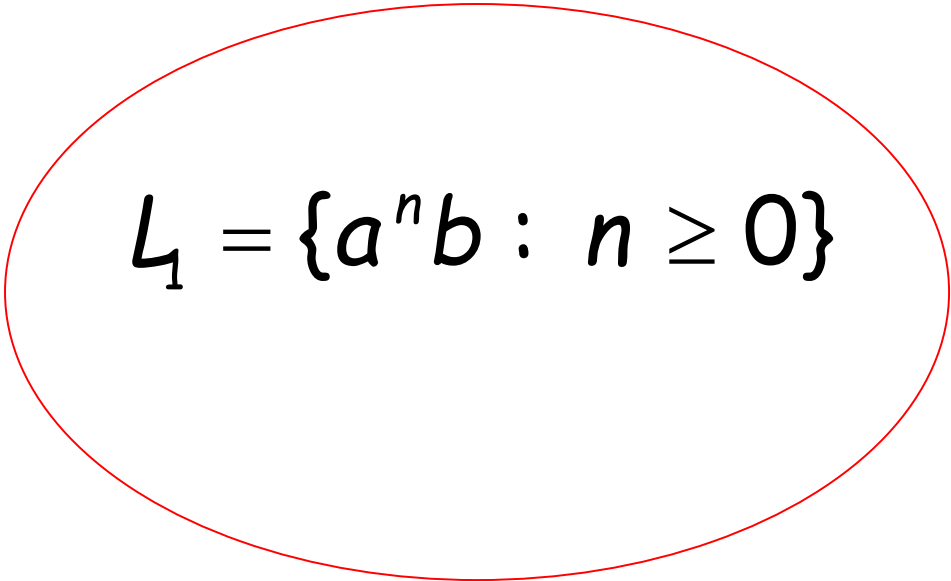
All Languages decidable by a  
deterministic Turing Machine  
in time  $O(T(n))$



Example:  $L_1 = \{a^n b : n \geq 0\}$

This can be decided in  $O(n)$  time

*TIME*( $n$ )


$$L_1 = \{a^n b : n \geq 0\}$$



# Other example problems in the same class

*TIME*( $n$ )

$$L_1 = \{a^n b : n \geq 0\}$$

$$\{ab^n aba : n, k \geq 0\}$$

$$\{b^n : n \text{ is even}\}$$

$$\{b^n : n = 3k\}$$

## Examples in class:

*TIME* ( $n^2$ )

$$\{a^n b^n : n \geq 0\}$$

$$\{ww^R : w \in \{a,b\}^*\}$$

$$\{ww : w \in \{a,b\}^*\}$$

Examples in class:

$TIME(n^3)$

CYK algorithm

$L_2 = \{\langle G, w \rangle : w \text{ is generated by context-free grammar } G\}$

Matrix multiplication

$L_3 = \{\langle M_1, M_2, M_3 \rangle : n \times n \text{ matrices and } M_1 \times M_2 = M_3\}$

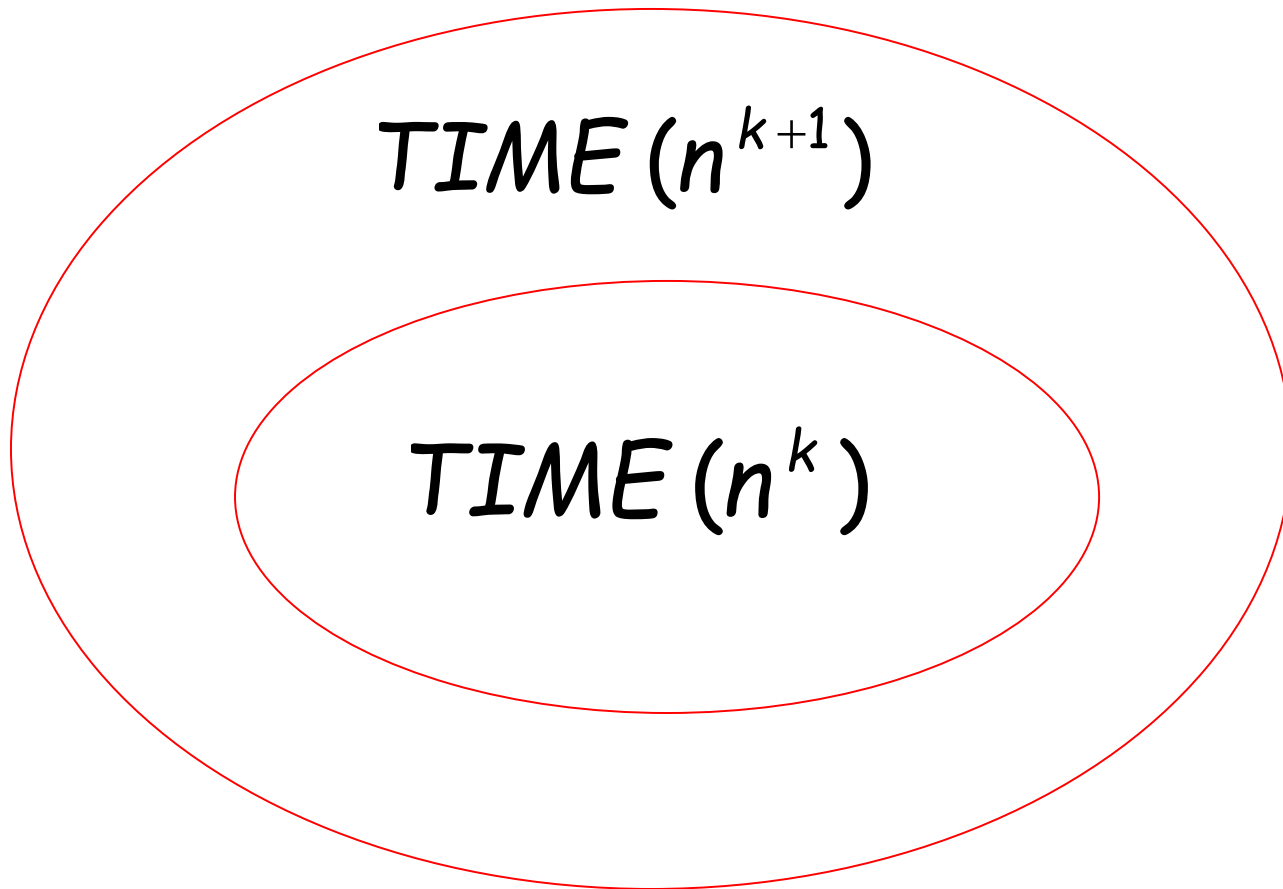
Polynomial time algorithms:  $TIME(n^k)$

constant  $k > 0$

Represents tractable algorithms:

for small  $k$  we can decide  
the result fast

It can be shown:  $TIME(n^k) \subset TIME(n^{k+1})$



# The Time Complexity Class $P$

$$P = \bigcup_k TIME(n^k)$$

Represents:

- polynomial time algorithms
- “tractable” problems

Class  $P$

$\{a^n b\}$

$\{a^n b^n\}$

$\{ww\}$

CYK-algorithm

Matrix multiplication

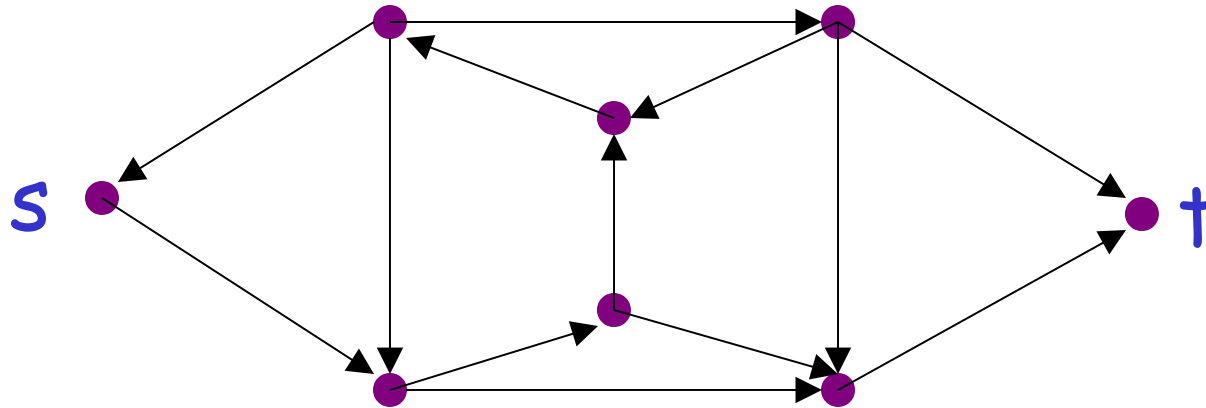
Exponential time algorithms:  $TIME(2^{n^k})$

Represent intractable algorithms:

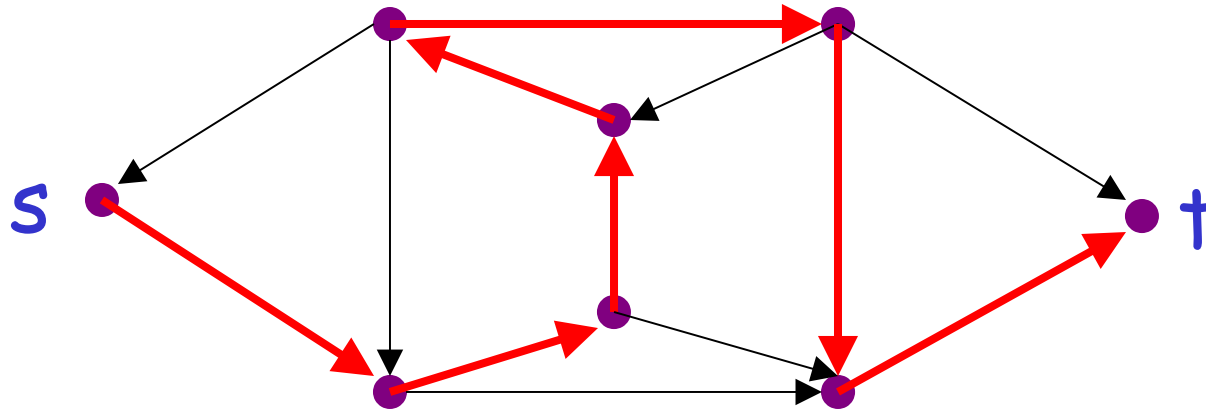
Some problem instances  
may take centuries to solve



# Example: the Hamiltonian Path Problem



Question: is there a Hamiltonian path from  $s$  to  $t$ ?



YES!

A solution: search exhaustively all paths

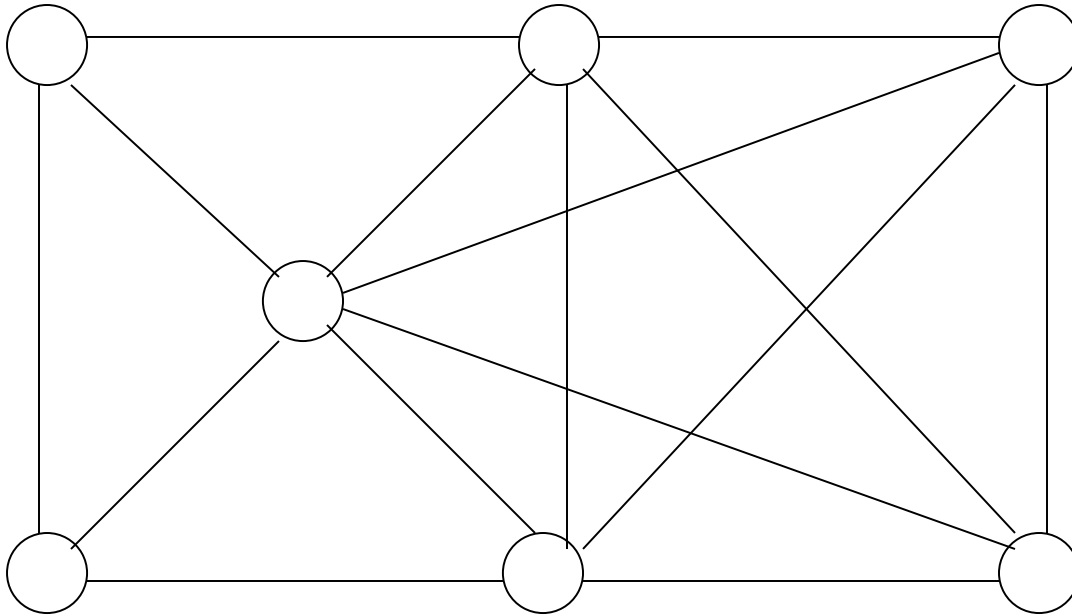
$L = \{ \langle G, s, t \rangle : \text{there is a Hamiltonian path in } G \text{ from } s \text{ to } t \}$

$$L \in TIME(n!) \approx TIME(2^{n^k})$$

Exponential time

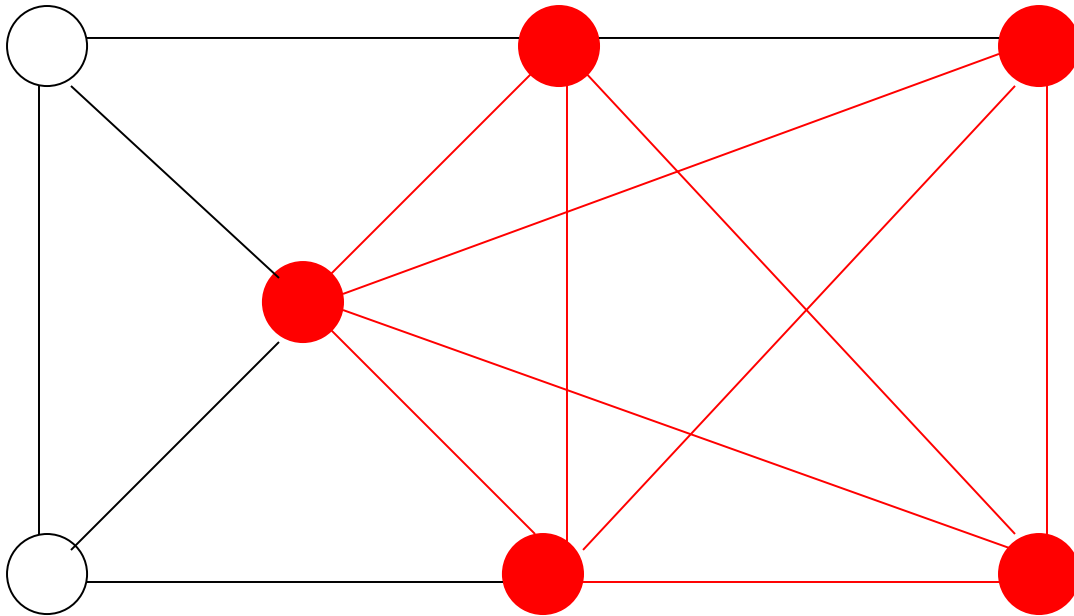
Intractable problem

# The clique problem



Does there exist a clique of size  $k$ ?

# The clique problem



$$k = 5$$

Does there exist a clique of size  $k$ ?

# Example: The Satisfiability Problem

Boolean expressions in  
Conjunctive Normal Form:

$$t_1 \wedge t_2 \wedge t_3 \wedge \cdots \wedge t_k \quad \text{clauses}$$

$$t_i = x_1 \vee \bar{x}_2 \vee x_3 \vee \cdots \vee \bar{x}_p$$

Variables

Question: is the expression satisfiable?

Example:  $(\bar{x}_1 \vee x_2) \wedge (x_1 \vee x_3)$

Satisfiable:  $x_1 = 0, x_2 = 1, x_3 = 1$

$$(\bar{x}_1 \vee x_2) \wedge (x_1 \vee x_3) = 1$$

Example:  $(x_1 \vee x_2) \wedge \bar{x}_1 \wedge \bar{x}_2$

Not satisfiable



$L = \{w : \text{expression } w \text{ is satisfiable}\}$

$L \in TIME(2^{n^k})$

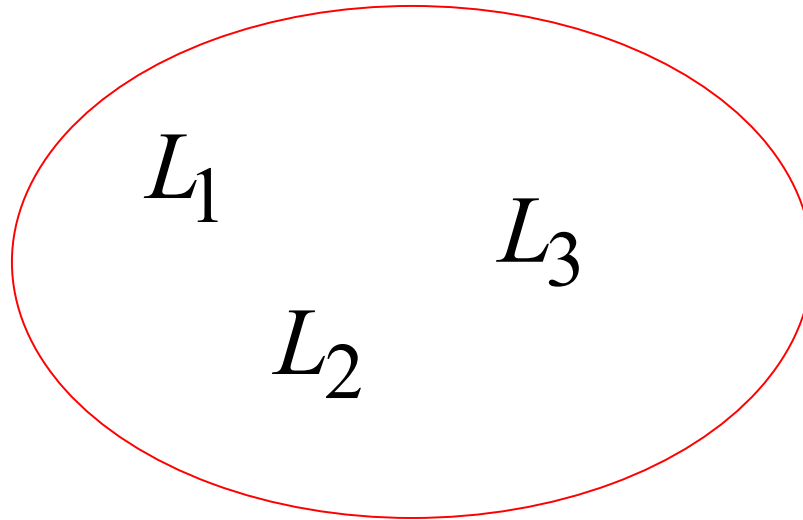
exponential

Algorithm:

search exhaustively all possible  
binary values of the variables

# Non-Determinism

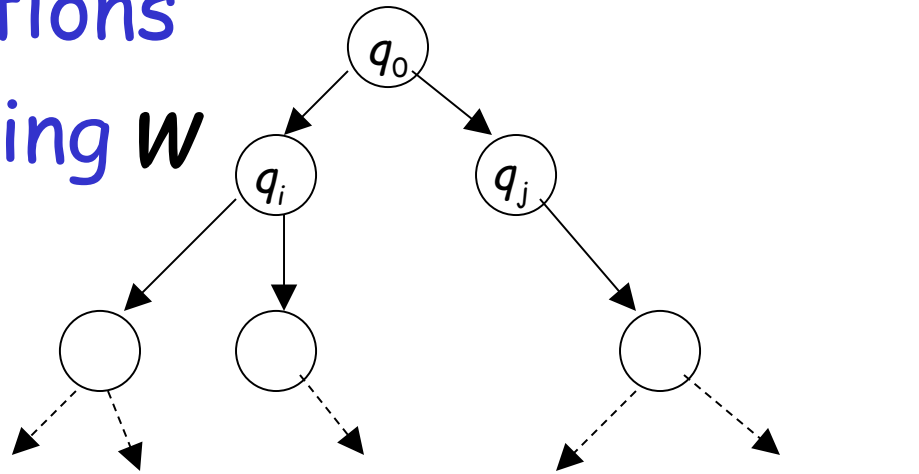
Language class:  $NTIME(T(n))$



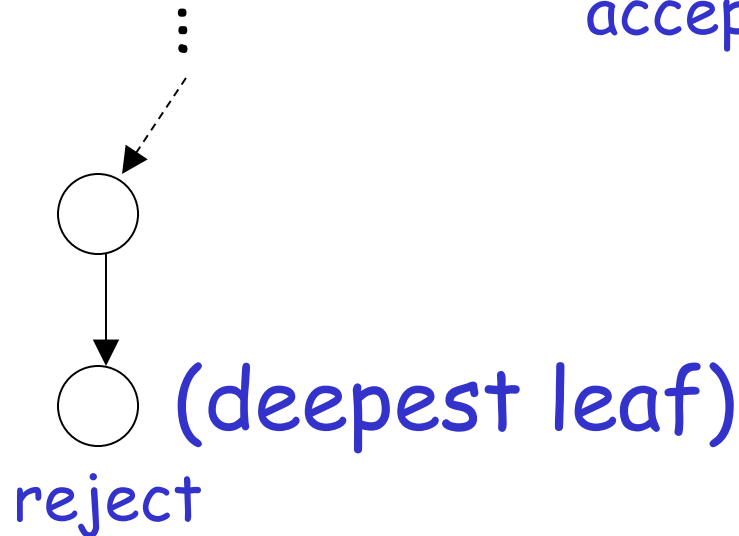
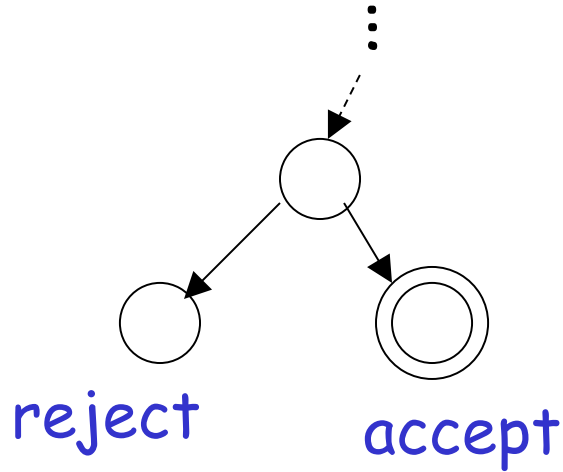
A Non-Deterministic Turing Machine  
decides each string of length  $n$   
in time  $O(T(n))$

All computations  
of  $M$  on string  $w$

$|w| = n$



depth  
 $T(n)$



# Non-Deterministic Polynomial time algorithms:

$$L \in NTIME(n^k)$$

# The class $NP$

$$NP = \bigcup_k NTIME(n^k)$$

Non-Deterministic Polynomial time

Example:    The satisfiability problem

$$L = \{w : \text{expression } w \text{ is satisfiable}\}$$

Non-Deterministic algorithm:

- Guess an assignment of the variables
- Check if this is a satisfying assignment

$$L = \{w : \text{expression } w \text{ is satisfiable}\}$$

Time for  $n$  variables:

- Guess an assignment of the variables  $O(n)$
- Check if this is a satisfying assignment  $O(n)$

Total time:  $O(n)$

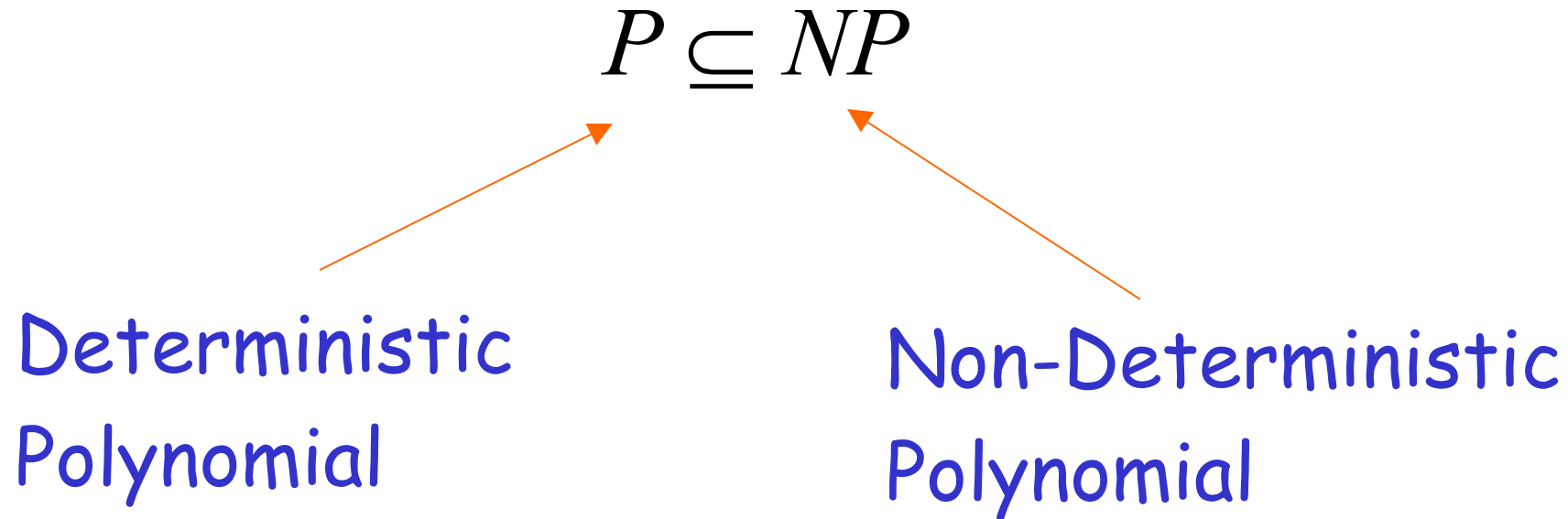
$$L = \{w : \text{expression } w \text{ is satisfiable}\}$$

$$L \in NP$$

The satisfiability problem is a  $NP$  - Problem



# Observation:



Open Problem:  $P = NP$  ?

WE DO NOT KNOW THE ANSWER

Open Problem:  $P = NP$  ?

Example: Does the Satisfiability problem have a polynomial time deterministic algorithm?

WE DO NOT KNOW THE ANSWER