# CSE211 – Formal Languages and Automata Theory

## FLAT_U4L12_DTIME and NTIME classes for Formal Language

**Dr. P. Saravanan**
School of Computing
SASTRA Deemed to be University

# Complexity

- A problem is decidable if there is an algorithm
- How to measure the complexity of program
- The set P is the set of problems or languages that can be decided by a Turing machine or some model of computation in polynomial time

Time Complexity: The number of steps during a computation

Space Complexity: Space used during a computation

# What we use

- Henceforth, we only consider decidable languages and deciders.
- Our computational model is a Turing Machine.
- Time: the number of computation steps a TM machine makes to decide on an input of size n.
- Space: the maximum number of tape cells a TM machine takes to decide on a input of size n.

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

For string of length $n$

time needed for acceptance: $O(n)$

# Complexity Classes

### Definition – Time Complexity Class TIME($t(n)$)

Let $t : \mathcal{N} \longrightarrow \mathcal{R}^+$ be a function.
$\text{TIME}(t(n)) = \{L(M) \mid M \text{ is a decider running in time } O(t(n))\}$

- TIME($t(n)$) is the class (collection) of languages that are decidable by TMs, running in time $O(t(n))$.
- $\text{TIME}(n) \subset \text{TIME}(n^2) \subset \text{TIME}(n^3) \subset \ldots \subset \text{TIME}(2^n) \subset \ldots$
- Examples:
  - $\{0^k 1^k \mid k \geq 0\} \in \text{TIME}(n^2)$
  - $\{0^k 1^k \mid k \geq 0\} \in \text{TIME}(n \log n)$ (next slide)
  - $\{w\#w \mid w \in \{0,1\}^*\} \in \text{TIME}(n^2)$

Language class: $DTIME(n)$

$DTIME(n)$

$L_1$ $L_2$ $L_3$

A Deterministic Turing Machine accepts each string of length $n$ in time $O(n)$

$$DTIME(n)$$

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

$$\{ww\}$$

In a similar way we define the class

$$DTIME(T(n))$$

for any time function: $T(n)$

Examples: $DTIME(n^2), DTIME(n^3), \ldots$

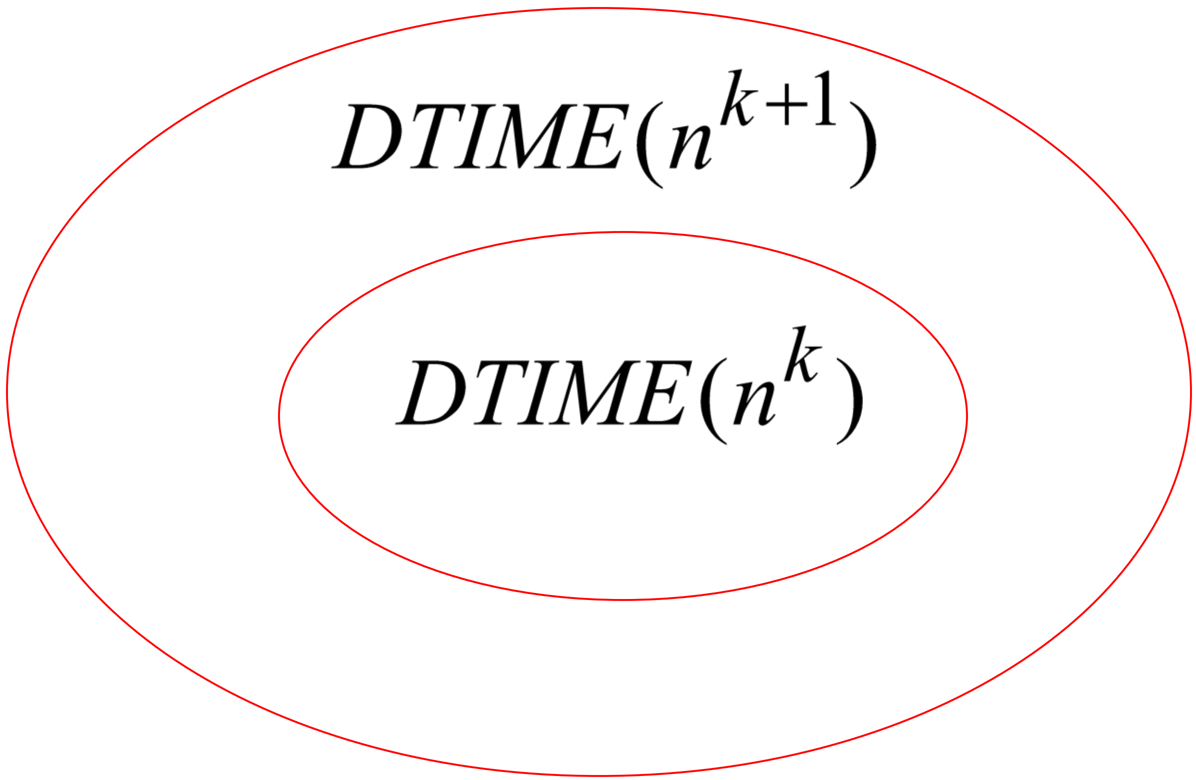
# Example: The membership problem for context free languages

$$L = \{w : w \text{ is generated by grammar } G\}$$

$$L \in DTIME(n^3)$$ (CYK - algorithm)

Polynomial time

**Theorem:** $DTIME(n^k) \subset DTIME(n^{k+1})$



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

Represent tractable algorithms:

For small $k$ we can compute the result fast

# NTIME

- In computational complexity theory, the complexity class NTIME(f(n)) is the set of decision problems that can be solved by a non-deterministic Turing machine
- It  runs in time O(f(n)). Here O is the big O notation, f is some function, and n is the size of the input (for which the problem is to be decided).

# NTIME

This means that there is a non-deterministic machine which, for

- a given input of size $n$,
- will run in time $O(f(n))$ (i.e. within a constant multiple of $f(n)$, for $n$ greater than some value), and
- will always "reject" the input if the answer to the decision problem is "no" for that input,
- while if the answer is "yes" the machine will "accept" that input for at least one computation path

# Space constraints

The space available to the machine is not limited, although it cannot exceed $O(f(n))$, because the time available limits how much of the tape is reachable.