# CS310 : Automata Theory 2019

## Lecture 42: Efficiency in computation
## Classifying problems by their complexity

Instructor: S. Akshay

IITB, India

18-04-2019

# Recap: Turing machines and computability

1. Turing machines
   (i) Definition & variants
   (ii) Decidable and Turing recognizable languages
   (iii) Church-Turing Hypothesis

# Recap: Turing machines and computability

1. Turing machines
   (i) Definition & variants
   (ii) Decidable and Turing recognizable languages
   (iii) Church-Turing Hypothesis
2. Undecidability
   (i) A proof technique by diagonalization
   (ii) Via reductions
   (iii) Rice's theorem

# Recap: Turing machines and computability

1. Turing machines
   (i) Definition & variants
   (ii) Decidable and Turing recognizable languages
   (iii) Church-Turing Hypothesis
2. Undecidability
   (i) A proof technique by diagonalization
   (ii) Via reductions
   (iii) Rice's theorem
3. Applications: showing (un)decidability of other problems
   (i) A string matching problem: Post's Correspondance Problem
   (ii) A problem for compilers: Unambiguity of Context-free languages
   (iii) Between TM and PDA: Linear Bounded Automata

# Recap: Turing machines and computability

1. Turing machines
   (i) Definition & variants
   (ii) Decidable and Turing recognizable languages
   (iii) Church-Turing Hypothesis
2. Undecidability
   (i) A proof technique by diagonalization
   (ii) Via reductions
   (iii) Rice's theorem
3. Applications: showing (un)decidability of other problems
   (i) A string matching problem: Post's Correspondance Problem
   (ii) A problem for compilers: Unambiguity of Context-free languages
   (iii) Between TM and PDA: Linear Bounded Automata
4. Efficiency in computation: time and space complexity.
   (i) Running time complexity: polynomial and exponential time
   (ii) Nondeterministic polynomial time, and the $P$ vs $NP$ problem.
   (iii) NP-completeness, the Cook-Levin Theorem.
   (iv) Space complexity classes

# Space complexity

Space complexity of a TM $M$ is the function $f : \mathbb{N} \rightarrow \mathbb{N}$, such that on any input of length $n$, $M$ scans atmost $f(n)$ many tape cells.

- We say, $M$ runs in $f(n)$ space.

# Space complexity

Space complexity of a TM $M$ is the function $f : \mathbb{N} \rightarrow \mathbb{N}$, such that on any input of length $n$, $M$ scans atmost $f(n)$ many tape cells.

- We say, $M$ runs in $f(n)$ space.
- If $M$ is an NTM, then its space complexity $f(n)$, is the max no. of tape cells scanned on any branch.

# Space complexity

Space complexity of a TM $M$ is the function $f : \mathbb{N} \to \mathbb{N}$, such that on any input of length $n$, $M$ scans atmost $f(n)$ many tape cells.

- We say, $M$ runs in $f(n)$ space.
- If $M$ is an NTM, then its space complexity $f(n)$, is the max no. of tape cells scanned on any branch.

## Space complexity classes

- $SPACE(f(n))$ is set of languages that can be decided by a $O(f(n))$ space DTM.

# Space complexity

Space complexity of a TM $M$ is the function $f : \mathbb{N} \to \mathbb{N}$, such that on any input of length $n$, $M$ scans atmost $f(n)$ many tape cells.

- We say, $M$ runs in $f(n)$ space.
- If $M$ is an NTM, then its space complexity $f(n)$, is the max no. of tape cells scanned on any branch.

## Space complexity classes

- $SPACE(f(n))$ is set of languages that can be decided by a $O(f(n))$ space DTM.
- $PSPACE$ is set of languages/problems that can be decided by a polynomial space DTM. $PSPACE = \bigcup_k SPACE(n^k)$.

# Space complexity

- SAT: $\exists x \exists y \varphi(x, y)$

# Space complexity

- SAT: $\exists x \exists y \varphi(x, y)$
- $\exists - \forall SAT$ : $\exists x \forall y \varphi(x, y)$

# Space complexity

- SAT: $\exists x \exists y \varphi(x, y)$
- $\exists - \forall SAT$ : $\exists x \forall y \varphi(x, y)$
- $\forall - \exists SAT$ : $\forall x \exists y \varphi(x, y)$

# Space complexity

- SAT: $\exists x \exists y \varphi(x, y)$
- $\exists - \forall SAT$ : $\exists x \forall y \varphi(x, y)$
- $\forall - \exists SAT$ : $\forall x \exists y \varphi(x, y)$

## Quantified Boolean Formula (QBF)

$\exists x_1 \forall x_2 \exists x_3 \forall x_4 \ldots Q x_n \varphi(x_1, \ldots, x_n)$

# Space complexity

- SAT: $\exists x \exists y \varphi(x, y)$
- $\exists - \forall SAT$ : $\exists x \forall y \varphi(x, y)$
- $\forall - \exists SAT$ : $\forall x \exists y \varphi(x, y)$

## Quantified Boolean Formula (QBF)

$\exists x_1 \forall x_2 \exists x_3 \forall x_4 \ldots Q x_n \varphi(x_1, \ldots, x_n)$

Formulate it as a game! Many more game examples are in PSPACE!

# Space complexity classes

- $SPACE(f(n))$ is set of languages that can be decided by a $O(f(n))$ space DTM.

# Space complexity classes

- $SPACE(f(n))$ is set of languages that can be decided by a $O(f(n))$ space DTM.
- $PSPACE$ is set of languages/problems that can be decided by a polynomial space DTM. $PSPACE = \bigcup_k SPACE(n^k)$.

## Exercises!

1. Define $NPSPACE$.

# Space complexity classes

- $SPACE(f(n))$ is set of languages that can be decided by a $O(f(n))$ space DTM.
- $PSPACE$ is set of languages/problems that can be decided by a polynomial space DTM. $PSPACE = \bigcup_k SPACE(n^k)$.

## Exercises!

1. Define $NPSPACE$.
2. What is the relation between
   2.1 $P$ and $PSPACE$? $NP$ and $NPSPACE$?
   2.2 $PSPACE$ and $EXPTIME$?
   2.3 $PSPACE$ and $NPSPACE$?

# Space complexity classes

- $SPACE(f(n))$ is set of languages that can be decided by a $O(f(n))$ space DTM.
- $PSPACE$ is set of languages/problems that can be decided by a polynomial space DTM. $PSPACE = \bigcup_k SPACE(n^k)$.

## Exercises!

1. Define $NPSPACE$.
2. What is the relation between
   2.1 $P$ and $PSPACE$? $NP$ and $NPSPACE$?
   2.2 $PSPACE$ and $EXPTIME$?
   2.3 $PSPACE$ and $NPSPACE$?

## Savitch's theorem

$NSPACE(f(n)) \subseteq SPACE(f^2(n))$, for $f(n) \geq \log n$

# Space complexity classes

- $SPACE(f(n))$ is set of languages that can be decided by a $O(f(n))$ space DTM.
- $PSPACE$ is set of languages/problems that can be decided by a polynomial space DTM. $PSPACE = \bigcup_k SPACE(n^k)$.

## Exercises!

1. Define $NPSPACE$.
2. What is the relation between
   2.1 $P$ and $PSPACE$? $NP$ and $NPSPACE$?
   2.2 $PSPACE$ and $EXPTIME$?
   2.3 $PSPACE$ and $NPSPACE$?

## Savitch's theorem

$NSPACE(f(n)) \subseteq SPACE(f^2(n))$, for $f(n) \geq \log n$

$$P \subseteq NP \subseteq PSPACE = NPSPACE \subseteq EXPTIME$$

End of Syllabus!