Department of CSE(H)

THEORY OF COMPUTATION
23CS2014

Topic:

PSPACE

AIM OF THE SESSION

To provide participants with a comprehensive understanding of PSPACE complexity class, its significance in computational theory, and its applications in solving problems that require polynomial space.

INSTRUCTIONAL OBJECTIVES

•Understanding PSPACE: Participants will be able to define the PSPACE complexity class and differentiate it from other complexity classes, such as P and NP.
•Problem Solving in PSPACE: Participants will be able to identify and analyze problems that belong to the PSPACE class, including strategies for developing algorithms that operate within polynomial space constraints.

LEARNING OUTCOMES

At the end of this session, you should be able to:

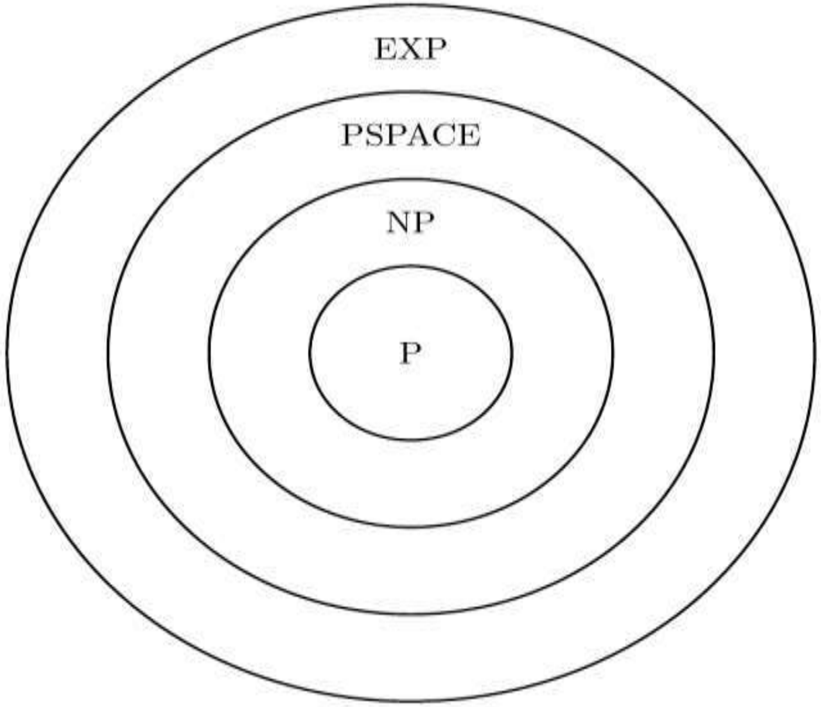
•Conceptual Knowledge: Participants will demonstrate an understanding of the fundamental properties of PSPACE and its relevance in computational theory.
•Application Skills: Participants will apply their knowledge of PSPACE to design and implement algorithms that efficiently solve problems within this complexity class, using appropriate data structures and methods to manage space constraints.

What is PSPACE?

- PSPACE consists of decision problems that can be solved using polynomial space (n^k for some constant k) on a deterministic or non-deterministic Turing machine.

PSPACE vs. Other Complexity Classes

- PSPACE includes problems solvable with polynomial space, unlike P, NP, and EXP. All P problems are in PSPACE, but PSPACE contains problems outside P. PSPACE is also a subset of EXP.



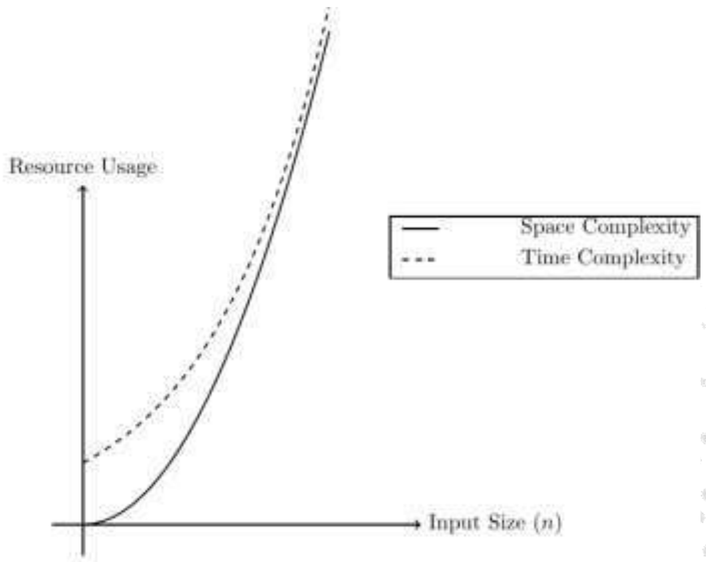
Space Complexity

- Space complexity refers to the amount of memory used by an algorithm. PSPACE focuses on problems solvable with polynomial space, even if the time required grows exponentially.

- Consider the True Quantified Boolean Formula (TQBF) problem, which is PSPACE-complete. The space complexity of solving this problem can be polynomial, but the time complexity may grow exponentially.

  Space Complexity : $O(n^k)$

  Time Complexity : $O(2^n)$



PSPACE-Complete Problems

- PSPACE-complete problems are the hardest problems in PSPACE. If any PSPACE-complete problem can be solved in polynomial space, then all problems in PSPACE can be solved in polynomial space.

Savitch's Theorem

- Savitch's theorem shows that NPSPACE equals PSPACE, meaning that non-deterministic polynomial space problems can be solved deterministically in polynomial space.

- If a problem can be solved in **non-deterministic** space $S(n)$, then it can also be solved in **deterministic** space $S(n)^2$ .Formally:

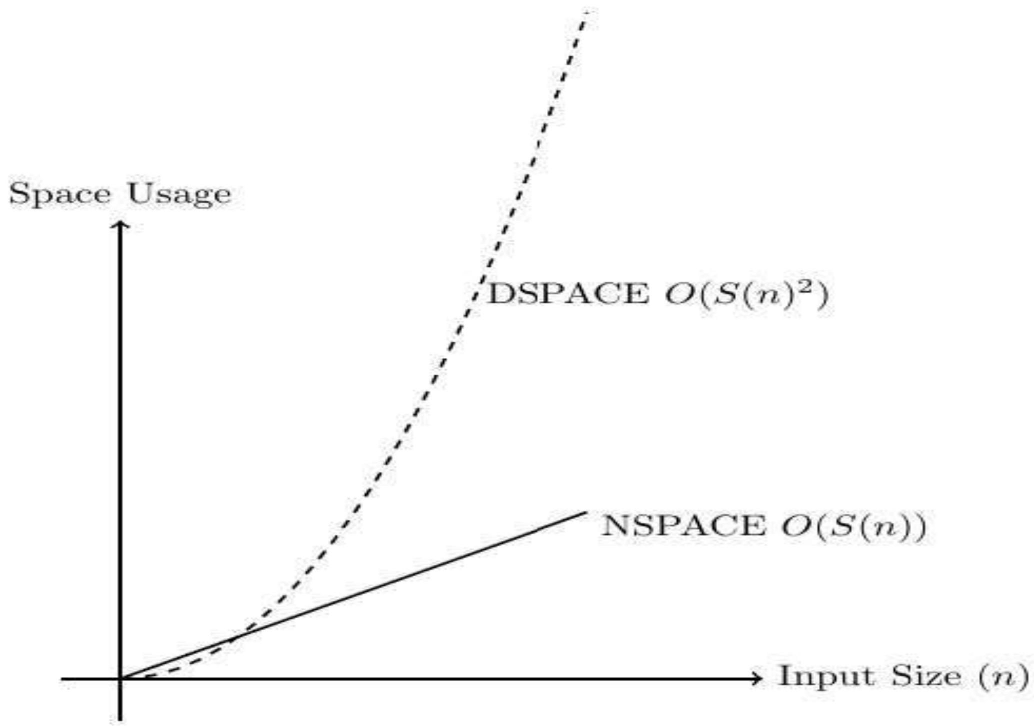
$$NSPACE\big(S(n)\big) \subseteq DSPACE\left(S(n)^2\right)$$

- $NSPACE\big(S(n)\big)$:The class of problems that can be solved by a non-deterministic TM using $S(n)$ space.
- $DSPACE\left(S(n)^2\right)$:The class of problems that can be solved by a deterministic TM using $S(n)^2$ space.

Savitch's Theorem: Example

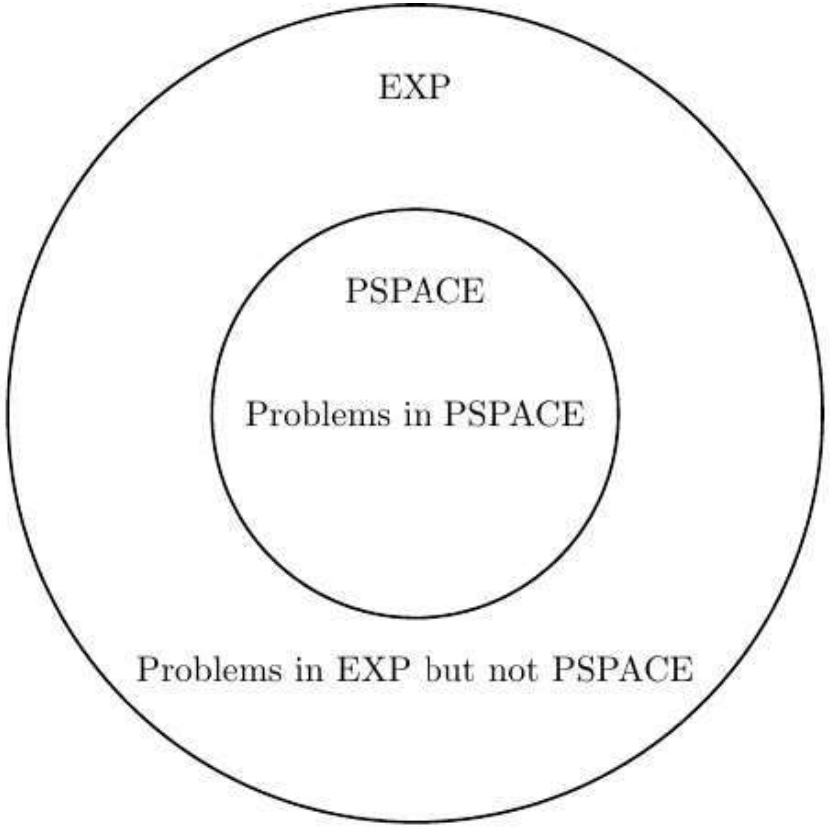
- Consider the problem of determining whether a directed graph has a path between two vertices, known as the **ST-Connectivity Problem**. For a graph with n nodes:
- Non-deterministic space: $O(\log n)$
- Deterministic Space : $O(log^2 n)$

  (Using Savitch’s Theorem)



PSPACE and EXP

- PSPACE is a subset of EXP, meaning that any problem solvable in PSPACE can also be solved in exponential time, though not all EXP problems are solvable in PSPACE e.g. **EXACT Circuit Satisfiability** problem.



PSPACE and EXP

- **EXACT Circuit Satisfiability** problem: The problem asks whether there is exactly one assignment of inputs to a Boolean circuit such that the output is true.
- Solving this problem requires storing exponentially many potential assignments (since each configuration must be checked to ensure that exactly one input satisfies the circuit). This demands **exponential space**.

THANK YOU

(DEEMED TO BE UNIVERSITY)

Team - TOC