Experiment #	<TO BE FILLED BY STUDENT>	Student ID	<TO BE FILLED BY STUDENT>
Date	<TO BE FILLED BY STUDENT>	Student Name	<TO BE FILLED BY STUDENT>

TUTORIAL SESSION 24:

Reducibility

Concept Building

Reducibility

It is a fundamental concept in the Theory of Computation that deals with the relationship between computational problems. It involves transforming one problem into another in such a way that solving the new problem provides a solution to the original problem. This transformation helps in understanding the complexity of different problems and classifying them based on their computational difficulty.

Key Concepts of Reducibility:

Transformation of Problems:

Reducibility involves creating a procedure to convert instances of one problem (Problem A) into instances of another problem (Problem B). If such a procedure exists, Problem A is said to be reducible to Problem B.

Preserving Solutions:

The transformation must preserve the solution's correctness. This means that a solution to the transformed instance of Problem B should directly correspond to a solution to the original instance of Problem A.

Polynomial-Time Reduction:

A common form of reducibility is polynomial-time reduction, where the transformation procedure can be executed in polynomial time. This type of reduction is denoted as A≤pBA \leq_p BA≤p B, meaning Problem A is polynomial-time reducible to Problem B.

Classifying Problems:

Reducibility helps in classifying problems into complexity classes, such as P, NP, and NP-complete. By showing that a problem can be reduced to an NP-complete problem, we can infer that the problem is at least as hard as the NP-complete problem.

Proving NP-Completeness:

To prove a new problem is NP-complete, it must be shown that the problem is in NP and that a known NP-complete problem can be reduced to it in polynomial time. This demonstrates that the new problem is as difficult as any problem in NP.

Experiment #	<TO BE FILLED BY STUDENT>	Student ID	<TO BE FILLED BY STUDENT>
Date	<TO BE FILLED BY STUDENT>	Student Name	<TO BE FILLED BY STUDENT>

Importance:

Understanding Complexity:

Reducibility allows us to understand the relative complexity of different problems. If a problem can be reduced to a known difficult problem, it suggests that the original problem is also difficult.

Algorithm Design:

By reducing complex problems to simpler ones, we can leverage existing efficient algorithms to solve the original problems. This approach guides the development of new algorithms and optimization techniques.

Classifying Computational Problems:

Reducibility helps in classifying problems into complexity classes like P, NP, and NP-complete. This classification is crucial for understanding which problems can be solved efficiently and which are inherently difficult.

Proving Intractability:

Demonstrating that a problem is NP-complete using reducibility shows that the problem is unlikely to have a polynomial-time solution, guiding researchers away from seeking efficient algorithms for such problems and towards approximation or heuristic approaches.

Experiment #	<TO BE FILLED BY STUDENT>	Student ID	<TO BE FILLED BY STUDENT>
Date	<TO BE FILLED BY STUDENT>	Student Name	<TO BE FILLED BY STUDENT>

Pre-Tutorial (To be completed by student before attending tutorial session)

1. What is reducibility in the context of computational theory?

Solution:

In computational theory, *reducibility* is transforming one problem into another to determine solvability or complexity.

2. What does it mean if problem A is reducible to problem B?

Solution

If problem A is reducible to problem B, solving B provides a solution for A. This means B is at least as hard as A in terms of complexity.

3. Why is reducibility important in computational complexity?

Solution:

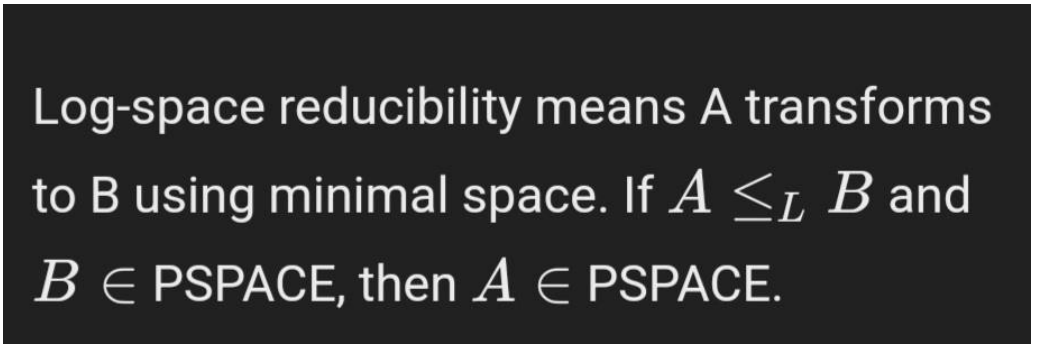
Reducibility helps classify problems by complexity, showing relationships between problems and identifying intractable or unsolvable problems.

Experiment #	<TO BE FILLED BY STUDENT>	Student ID	<TO BE FILLED BY STUDENT>
Date	<TO BE FILLED BY STUDENT>	Student Name	<TO BE FILLED BY STUDENT>

IN-TUTORIAL (To be carried out in presence of faculty in classroom)

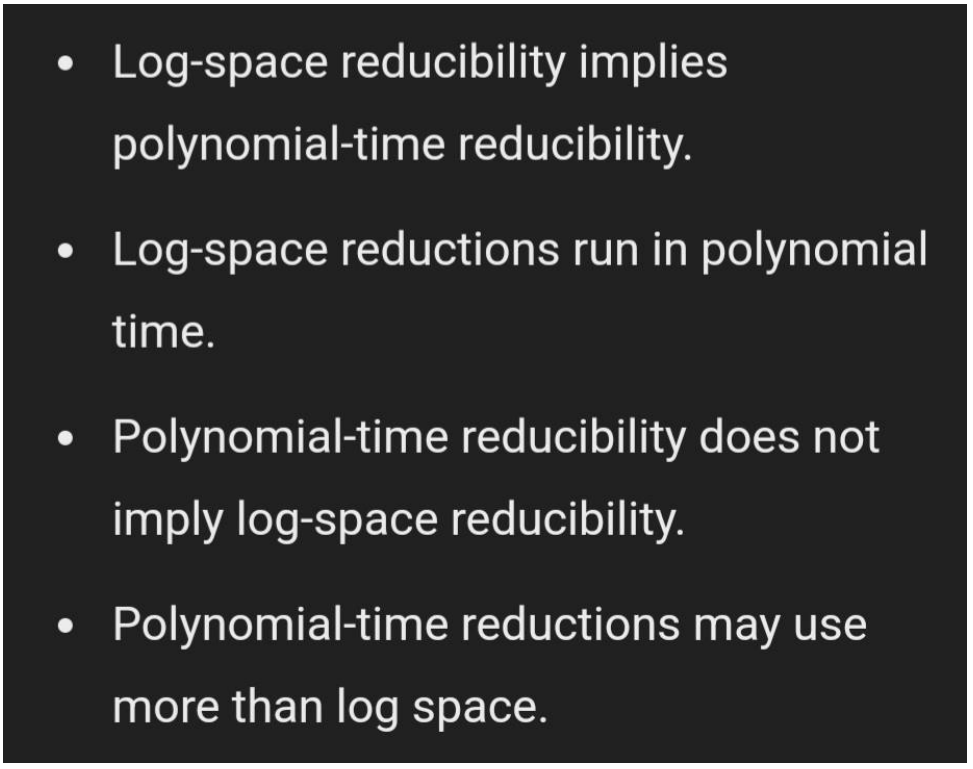
1. Explain the concept of log-space reducibility. If A ≤L B and B ∈ PSPACE, what can be concluded about A?

Solution



2. What is the relationship between log-space reducibility and polynomial-time reducibility? Does one imply the other?

Solution:



Experiment #	<TO BE FILLED BY STUDENT>	Student ID	<TO BE FILLED BY STUDENT>
Date	<TO BE FILLED BY STUDENT>	Student Name	<TO BE FILLED BY STUDENT>

3. Explain the concept of polynomial-time reducibility. If A ≤P B and B ∈ NP, what can be concluded about A?

Solution:

Polynomial-time reducibility means A transforms to B in polynomial time. If $A \leq_P B$ and $B \in$ NP, then $A \in$ NP.

4. Suppose A ≤L B and B ≤L C. What can be concluded about the relationship between A and C?

Solution:

If $A \leq_L B$ and $B \leq_L C$, then $A \leq_L C$. This means A is log-space reducible to C, as log-space reductions are transitive.

Experiment #	<TO BE FILLED BY STUDENT>	Student ID	<TO BE FILLED BY STUDENT>
Date	<TO BE FILLED BY STUDENT>	Student Name	<TO BE FILLED BY STUDENT>

Post-Tutorial (To be carried out by student after attending tutorial session)

1. If A ≤P B and B ∈ EXPTIME, what can be concluded about the complexity of A?

Solution:

If $A \leq_P B$ and $B \in$ EXPTIME, then $A \in$ EXPTIME. This is because a polynomial-time reduction from A to B implies A can be solved within the same time complexity as B, which is EXPTIME.

2. Define polynomial-time reducibility (≤P) and explain its significance in computational complexity.

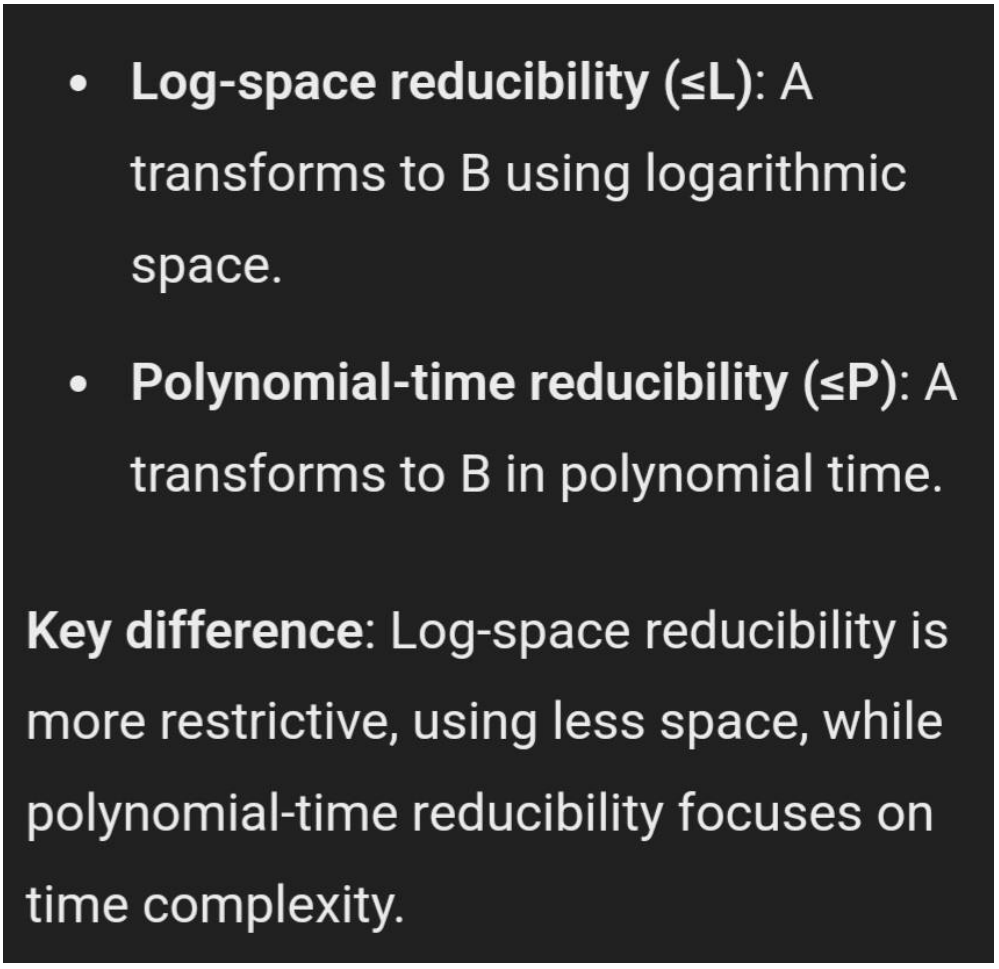
Solution

Polynomial-time reducibility (≤P): A transforms to B in polynomial time.
Significance: Helps classify problem difficulty and establish problem hardness.

Experiment #	<TO BE FILLED BY STUDENT>	Student ID	<TO BE FILLED BY STUDENT>
Date	<TO BE FILLED BY STUDENT>	Student Name	<TO BE FILLED BY STUDENT>

3. What is the difference between log-space reducibility (≤L) and polynomial-time reducibility (≤P)?

Solution:

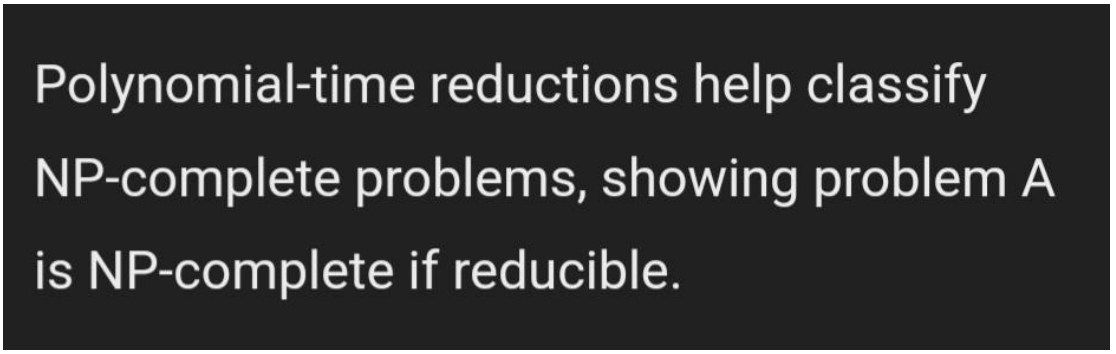


Experiment #	<TO BE FILLED BY STUDENT>	Student ID	<TO BE FILLED BY STUDENT>
Date	<TO BE FILLED BY STUDENT>	Student Name	<TO BE FILLED BY STUDENT>

Viva – Questions

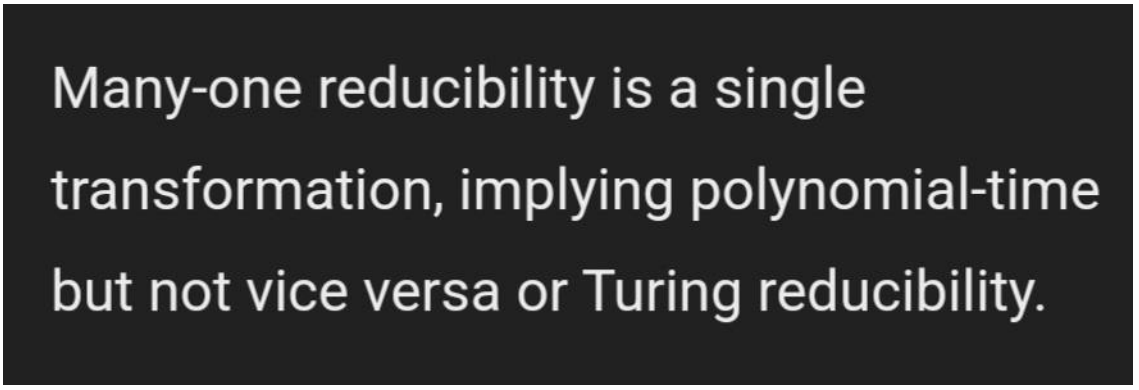
1. What is the significance of polynomial-time reductions in the context of NP-completeness?

Solution:



2. Explain the concept of many-one reducibility and its relationship to other types of reducibility.

Solution:



(For Evaluator's use only)

Comment of the Evaluator (if Any)	Evaluator's Observation Marks Secured: out of **50** Full Name of the Evaluator: Signature of the Evaluator Date of Evaluation: