Chapter 7

Computational Complexity and Interactability

Polynomial time algorithm

- A ***polynomial-time algorithm*** is one whose worst-case time complexity is bounded above by a polynomial function of its input size.

- Algorithms with the following worst-case time complexities are all polynomial time.

$$2n \qquad 3n^3 + 4n \qquad 5n + n^{10} \qquad n \lg n$$

- Algorithms with the following worst-case time complexities are not polynomial time

$$-2^n \qquad 2^{0.01n} \qquad 2^{\sqrt{n}} \qquad n!$$

Intractable Problem

- In computer science, a problem is called ***intractable*** if it is impossible to solve it with a polynomial-time algorithm.
- Intractability is a property of a problem; it is not a property of any one algorithm for that problem.
- For a problem to be intractable, there must be no polynomial-time algorithm that solves it.

Three general categories of problems

1. Problems for which polynomial-time algorithms have been found
2. Problems that have been proven to be intractable
3. Problems that have not been proven to be intractable, but for which polynomial-time algorithms have never been found

- most problems in computer science seem to fall into either the first or third category.

Problems That Have Not Been Proven to Be Intractable but for Which Polynomial-Time Algorithms Have Never Been Found

- the 0–1 Knapsack problem,
- the Travelling Salesperson problem,
- the *m*-Coloring problem for $m \geq 3$,
- the Hamiltonian Circuits problem
- …

Decision Problems

- The output of a decision problem is a simple "yes" or "no" answer.

Optimization Problems

Each optimization problem, however, has a corresponding decision problem

Example

- The ***Traveling Salesperson Optimization problem*** is to determine a tour with minimal total weight on its edges.

- The ***Traveling Salesperson Decision problem is*** to determine for a given positive number *d* whether there is a tour having total weight no greater than *d*.

Definition: P problems

- ***P*** is the set of all decision problems that can be solved by polynomial-time algorithms.

Nondeterministic Algorithm

Nondeterministic algorithm composed of the following two separate stages:

- **Guessing (Nondeterministic) Stage:**
- **Verification (Deterministic) Stage:**

In computer science, a **Nondeterministic algorithm** is an algorithm that, even for the same input, can exhibit different behaviors on different runs

Definition: NP problems

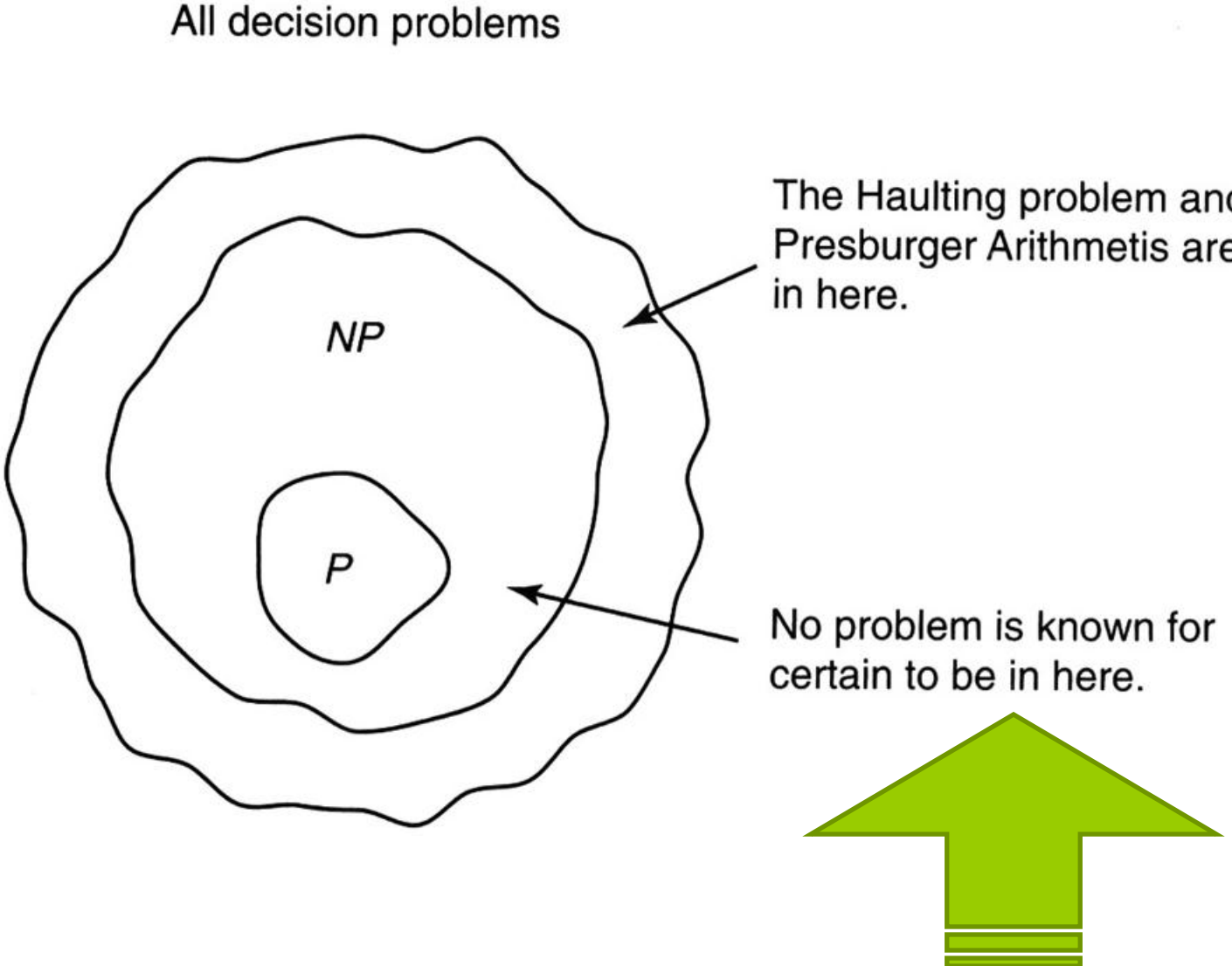
- The abbreviation **NP** refers to **nondeterministic polynomial time**.

- **NP** is the set of all decision problems whose verification stage is a polynomial-time algorithm

What decision problems are not in *NP*?

- The only decision problems that have been proven not to be in *NP* are the same ones that have been proven to be intractable.
  - The Halting problem
  - …

P & NP


All decision problems
NP
P
The Haulting problem and Presburger Arithmetis are in here.
No problem is known for certain to be in here.

NP-Complete

The problems which are all equivalent in the sense that if any one is in *P*, they all must be in *P.* Such problems are called

NP-complete.

Reducibility

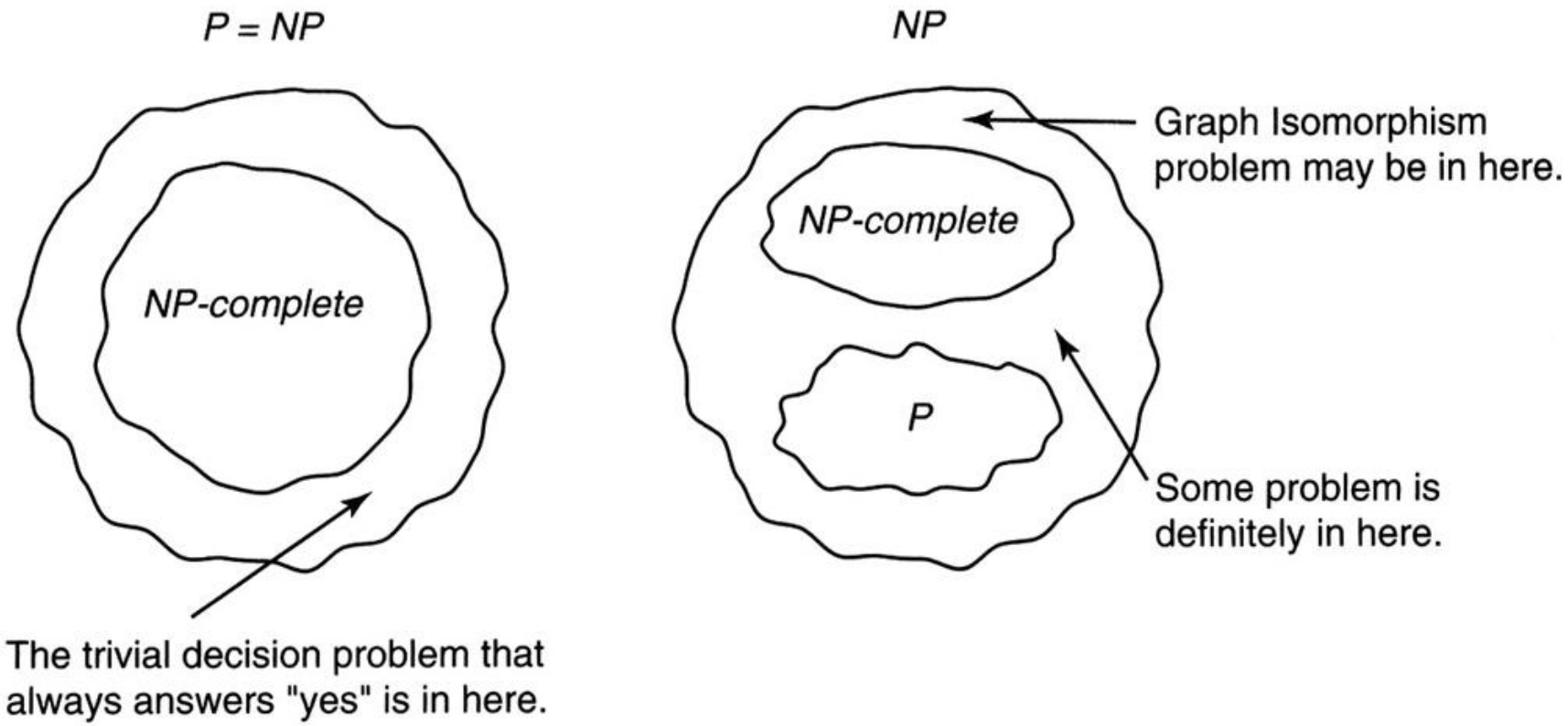
- Problem A reduces to problem B if there exists a polynomial-time transformation algorithm from decision problem *A* to decision problem *B*,
- In symbols, we write $A \propto B$

Theorem 9.3

A problem *C* is *NP*-complete if both of the following are true:

- C is in *NP.*
- For some other *NP*-complete problem *B:*

$$B \propto C$$


P = NP
NP-complete
The trivial decision problem that always answers "yes" is in here.
NP
NP-complete
P
Graph Isomorphism problem may be in here.
Some problem is definitely in here.

Solving NP-Complete Problems

NP-complete problems are often addressed by using heuristic methods and approximation algorithms.