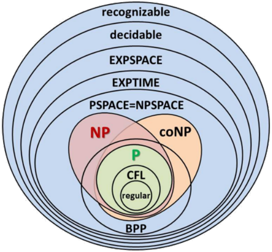
Reductions, P and NP

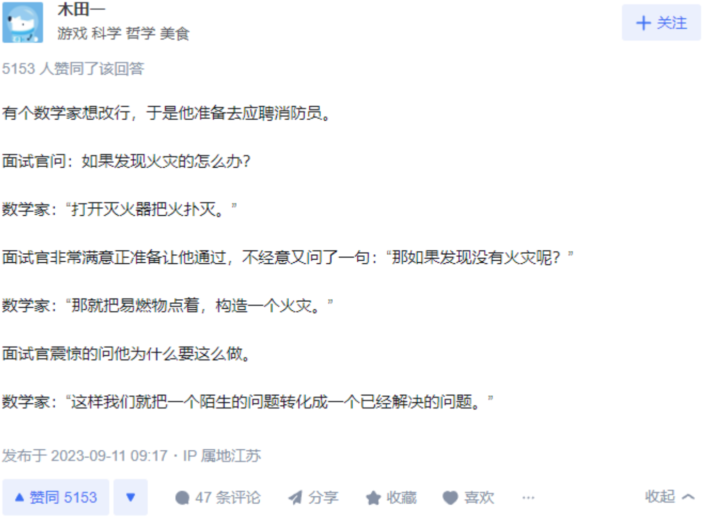
CS101 Fall 2024

CS101 Course Team

Dec 2024



What is reduction



木田一

游戏 科学 哲学 美食

5153 人赞同了该回答

有个数学家想改行，于是他准备去应聘消防员。

面试官问：如果发现火灾的怎么办?

数学家：“打开灭火器把火扑灭。”

面试官非常满意正准备让他通过，不经意又问了一句：“那如果发现没有火灾呢？”

数学家：“那就把易燃物点着，构造一个火灾。”

面试官震惊的问他为什么要这么做。

数学家：“这样我们就把一个陌生的问题转化成一个已经解决的问题。”

发布于 2023-09-11 09:17 · IP 属地江苏

What is reduction

A (Karp) reduction is a mapping that maps yes-instances of L to yes-instances of L' and no-instances of L to no-instances of L'.

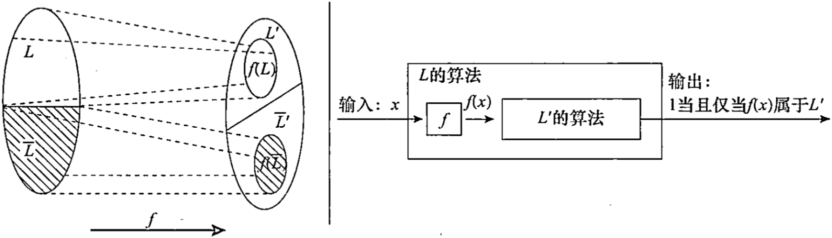
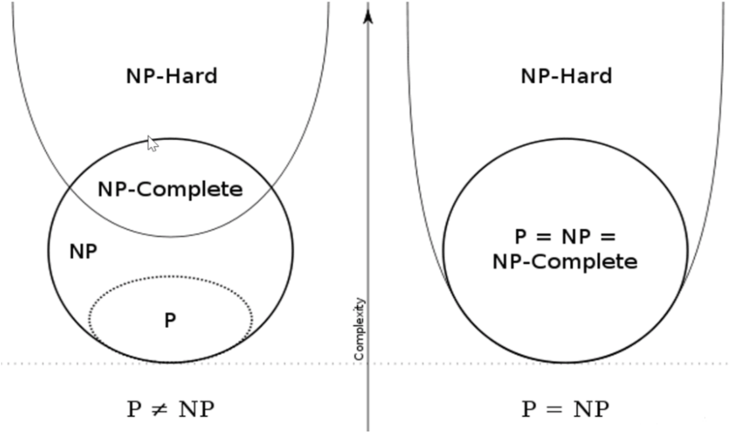


Figure: Karp reduction

Reduce from A to B: $A \leq_p B$, B is harder for A.

(Figure cite: Computational Complexity: A Modern Approach by Sanjeev Arora and Boaz Barak.)

Brief view of complexity classes



https://complexityzoo.uwaterloo.ca/Complexity_Zoo

P, and NP, and so on

Decision Problem

- A ‘problem’ (or more formally, language) L is defined as a set of strings in Σ^*. (where Σ is the ‘alphabet’.)
- Instance s is a string in Σ^*.
- Algorithm A decides L iff $A(x) = \text{yes} \Leftrightarrow x \in L$.

A runs in polynomial time if $\forall s$, $A(s)$ terminates in $\leq \mathit{poly}(|s|)$ steps.

P is the set of decision problems for which there exists a poly-time algorithm (on a deterministic Turing machine).

P, and NP, and so on

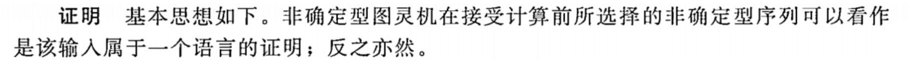
A formal definition of NP

NP is the set of decision problems for which there exists a poly-time algorithm (on a non-deterministic Turing machine).

Another way to define NP: can be verified in polynomial time and polynomial size.

- Certificate: Answer that needs to be verified.
- Certifier: Algorithm that verifies.

NP: Polynomial-time certifier and polynomial-size certificate.



证明　基本思想如下。非确定型图灵机在接受计算前所选择的非确定型序列可以看作是该输入属于一个语言的证明；反之亦然。

Figure: An intuitive understanding of two definitions

(I don't feel like typing. That's it.)

(Cite: Computational Complexity: A Modern Approach by Sanjeev Arora and Boaz Barak.)

P, and NP, and so on

Other types of problems

- Decision problem: Does there exist a vertex cover of size $\leq k$?
- Search problem: Find a vertex cover of size $\leq k$.
- Optimization problem: Find a vertex cover of minimum size.

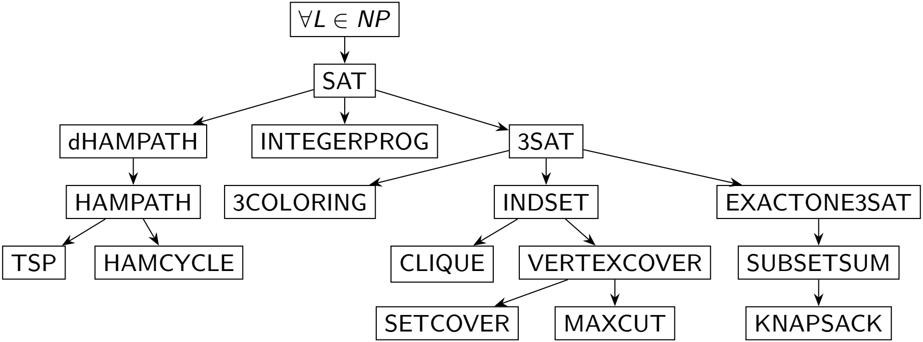
Other complexity classes

- EXP: Problems that take at most exponential amount of time.
- PSPACE: Problems that take polynomial amount of space.
- coNP: Problems whose no-instances can be verified in polynomial time.

$\mathrm{P} \subseteq \mathrm{NP} \subseteq \mathrm{PSPACE} \subseteq \mathrm{EXP}$, $\mathrm{P} \neq \mathrm{EXP}$, $\mathrm{P} \in \mathrm{NP} \cap co\mathrm{NP}$.

Brief view of NP-Complete Problems

A brief map of common NP-Complete problems:



$\forall L \in NP$

SAT

dHAMPATH

INTEGERPROG

3SAT

HAMPATH

3COLORING

INDSET

EXACTONE3SAT

TSP

HAMCYCLE

CLIQUE

VERTEXCOVER

SUBSETSUM

SETCOVER

MAXCUT

KNAPSACK

(Partially cite: Computational Complexity: A Modern Approach by Sanjeev Arora and Boaz Barak.)

Why KNAPSACK is not in P

Proof: Can be easily reduced from SUBSET-SUM.
But it exists an $O(nW)$ algorithm!

Pseudo polynomial-time algorithm

An algorithm is said to be pseudo-polynomial time if it takes polynomial time in the **value domain**.

What is the difference?
Input an integer x takes $O(\log|x|)$ time. That is, the length of input is $poly(\log W)$ for KNAPSACK, while it takes $poly(W)$ time.

How to prove...?

NP-Complete

1. Show that $Y \in \text{NP}$.
2. Choose an NP-Complete problem X.
3. Prove that $X \leq_p Y$. (Always by reduction)

P

Find a polynomial-time algorithm for it.

$\text{P} = \text{NP}$

A natural way: Find a polynomial-time algorithm for a NP-Complete Problem X.

Why we need reduction

By different kinds of reduction (and its transitivity), we can build equivalent classes i.e. complexity classes.

Reduction constructs a more easy and natural way to prove one's complexity lower bound. (While an efficient algorithm gives out an upper bound.)

However, the natural way to prove $\mathrm{P} = \mathrm{NP}$ is proved to be hardly existing.

One more thing...

If you want to know more about ‘complexity’ or other things about it:

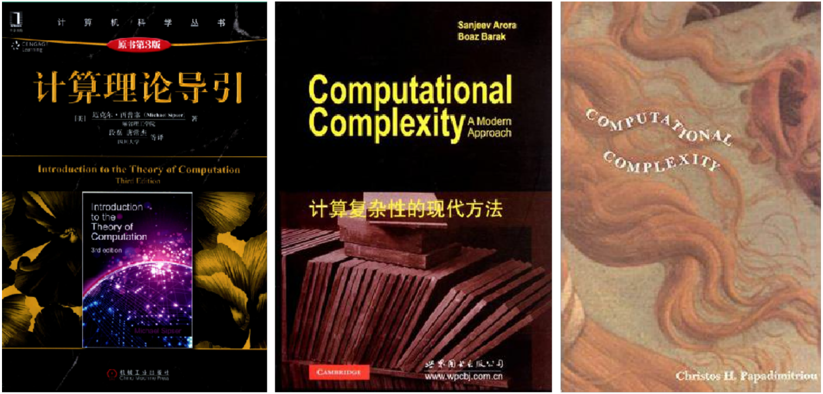


Figure: Recommend Reading