P, NP, and NP-Complete

Jidong Yuan
Beijing Jiaotong University
yuanjd@bjtu.edu.cn

SCC 120: Fundamentals of Computer Science

Complexity class P (Polynomial-Time)

Tractable decision problems

- A decision problem $D \in P$ iff any instance of D can be *decided* (solved) in polynomial time (by a deterministic Turing Machine (DTM))
    - Is $k \geq k'$? (where k, k' are both integers)
    - Is s a substring of s'?
    - Is the array sorted?
    - Does the list of names contain a given name?
    - Does a given name occur twice in the list?

Complexity class NP

Tractable verification problems

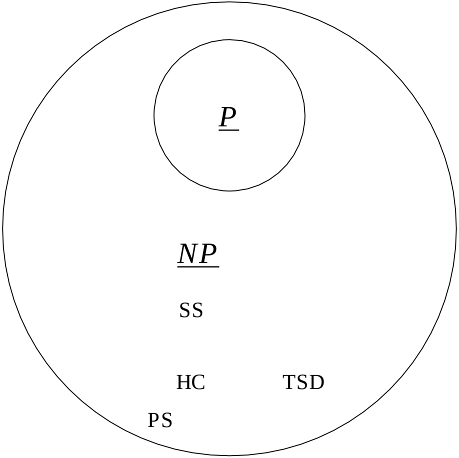
- A decision problem $D \in NP$ iff a *certificate* C for any instance of D can be *verified* (checked) in polynomial time (by a DTM)
  - Return true if C is actually a solution to D
- (Equivalently, a decision problem $D \in NP$ iff any instance of D can be *decided* in polynomial time by a nondeterministic TM.)
- Many decisions problems that are hard to solve are easy to verify
  - PS: Does $p = 1$, $q = 1$ satisfy $(p \wedge \neg p) \vee q)$?
  - PS: Does $p = 1$, $q = 0$ satisfy $(p \wedge \neg p) \vee q)$?
  - TSD: Is $(London, Bath, Edinburgh, Lancaster)$ a tour of at most 7?
  - TSD: Is $(London, Bath, Edinburgh, Lancaster, London)$ a tour of at most 5?
  - Subset sum: Does subset $\{-2, 2\}$ of $\{4, 2, -3, -2, 5\}$ sum to 0?

More Decision Problems

- All problems in P also in NP, that is, $P \subseteq NP$
  - Intuition: problems that are easy to decide are easy to verify
  - Example
    - Does "Sara" appear twice in $(Sara, Kapil, Sara, Julia, Rita, Ali)$?
    - Certificate: "Sara" at positions 0 and 2. Yes!
  - Example
    - Is "str" a substring of "adasasdstrasdas"?
    - Certificate: "str" begins at position 5. No! Not a valid certificate!
    - Certificate: "str" begins at position 7. Yes!

Picture So Far

P and *NP*



Polynomial-Time Algorithms for Hard Problems?

Open Question

- Not found yet
- But no proof that one cannot be found
- Question: If a decision problem is efficiently verifiable, then is it also efficiently decidable?
    - $P = NP$?
    - One of the Millenium Prize Problems (you get a million dollars for solving it)

NP-Complete Problems

Characterization of the hard problems in *NP*

- A decision problem $D \in NP$*-complete* iff
  - $D \in NP$, and
  - each problem in NP is reducible to D in polynomial time
- Say some problem $D \in NP$*-complete*
  - A polynomial-time algorithm for D implies a polynomial-time algorithm for every problem in NP

NP-Complete Problems

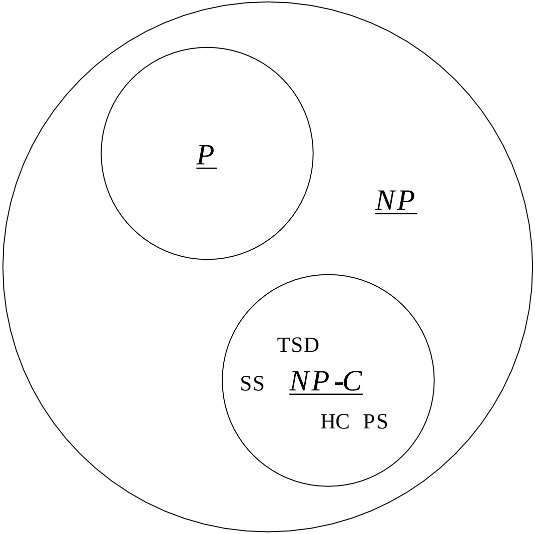
All our hard problems

- Propositional Satisfiability
- Subset sum
- Hamiltonian cycle
- Traveling salesman (decision)
- Many more …

If you find a polynomial algorithm for any of these, then you've found a polynomial algorithm for all problems in NP, including these

Complete Picture

Open question: $P = NP$?



Usefulness of Class *NP*-*Complete*

Showing other problems to be hard, that is, in *NP*-*Complete*

- To show a problem D is in *NP*-*Complete*
  - Show D is in *NP*
  - Give a polynomial-time reduction for some $D' \in$ *NP*-*Complete* to D
    - If you find a polynomial-time algorithm for D, then you have found a polynomial-time algorithm for D', which means you have found a polynomial-time algorithm for all problems in *NP*.