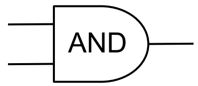
Intractable problems and the class NP

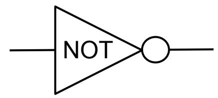
- Some problems seem to take a lot of time to solve by computer
  - Not because they have a lot of work to do (like sorting 10^9 integers), but for more fundamental reasons (like finding a path in a maze)
  - Algorithms exist but their temporal complexity is too high (perhaps all known algorithms are exponential)
- A major example is the class of NP problems
  - A problem is in the class NP if it is possible to *verify* a potential solution in polynomial time complexity
    - NP means Nondeterministic Polynomial time
    - Nondeterministic means that the algorithm can *choose* the right solution; it only has to verify that the solution is correct
  - But *finding* a solution from scratch seems to be much more expensive (exponential complexity)!

Satisfiability of digital circuits (and the question P=NP)

- Given a digital logic circuit without memory, built with And and Not gates. The circuit has *n* inputs and one output.
  - Is there a set of input values that makes the output true?



- This problem is in NP: it is easy to verify a potential solution

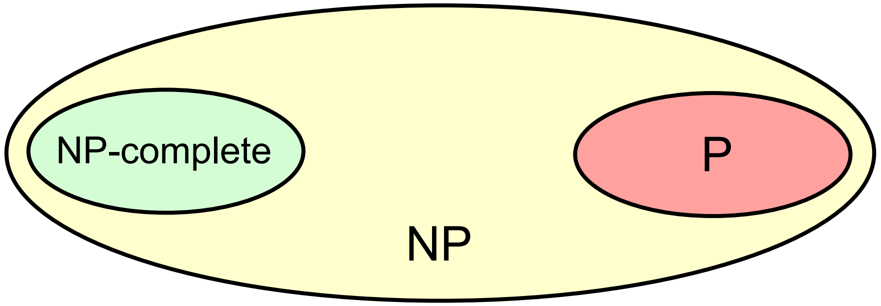


- But it is hard to find a solution
  - After several decades of work, no computer scientist or mathematician has found an algorithm that is substantially better (in the general case) than simply trying all 2^n possible inputs!
  - The best known algorithm has exponential time complexity
  - It is suspected that no polynomial time algorithm exists (but there is no proof)

- Eternal glory awaits the person who (1) proves that no polynomial time algorithm exists or (2) finds a polynomial time algorithm
  - The $1000000 question: Is P=NP? Is finding a solution as easy as verifying a solution? “Can creativity be automated?”
  - This is a deep question with many ramifications in philosophy and physical science. Some people think it should be accepted as a new physical law, in analogy to the Laws of Thermodynamics.

NP-complete problems

- Some problems in NP have the property that if an efficient algorithm can be found for it, then it is possible to derive an efficient algorithm for all problems in NP
- These problems are called NP-complete problems
  - They are the “hardest problems” in NP
- Satisfiability of digital circuits is an NP-complete problem



(Diagram assumes P≠NP)

Living with NP-complete problems

- NP-complete problems are often encountered in practice
- So how can we solve these problems, if the best known algorithm is exponential time?
- Sometimes it is possible to modify the problem to avoid the exponential cases
  - We can use an algorithm that gives an approximate answer or that sometimes gives no answer
  - For example, the Traveling Salesman Problem: what is the route of a traveler who visits all cities in an area so that the total distance traveled is minimal?
    - This is also an NP-complete problem
    - If we are happy with a distance that is within 10% of the minimal distance, then the algorithm is polynomial time