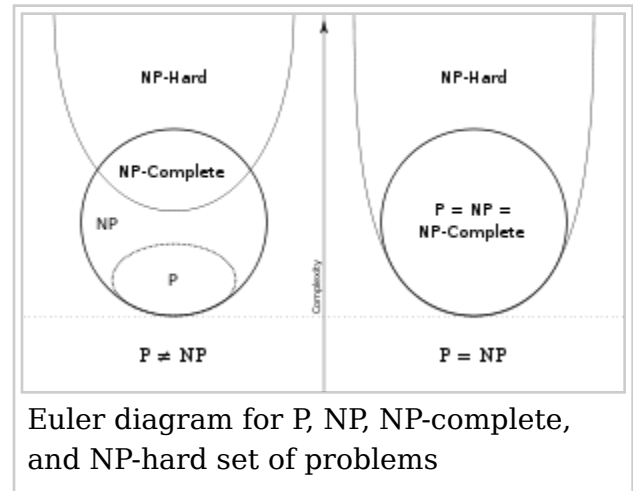


NP-hard

From Wikipedia, the free encyclopedia

NP-hard (Non-deterministic Polynomial-time hard), in computational complexity theory, is a class of problems that are, informally, "at least as hard as the hardest problems in NP". More precisely, a problem H is NP-hard when every problem L in NP can be reduced in polynomial time to H .^{[1]:80} As a consequence, finding a polynomial algorithm to solve any NP-hard problem would give polynomial algorithms for all the problems in NP, which is unlikely as many of them are considered hard.^[2]



A common mistake is to think that the *NP* in *NP-hard* stands for *non-polynomial*. Although it is widely suspected that there are no polynomial-time algorithms for NP-hard problems, this has never been proven. Moreover, the class NP also contains all problems which can be solved in polynomial time.

Contents

- 1 Definition
- 2 Consequences
- 3 Examples
- 4 NP-naming convention
- 5 Application areas
- 6 References

Definition

A decision problem H is NP-hard when for any problem L in NP, there is a polynomial-time reduction from L to H .^{[1]:80} An equivalent definition is to require that any problem L in NP can be solved in polynomial time by an oracle machine with an oracle for H .^[3] Informally, we can think of an algorithm that can call such an oracle machine as a subroutine for solving H , and solves L in polynomial time,

if the subroutine call takes only one step to compute.

Another definition is to require that there is a polynomial-time reduction from an NP-complete problem G to H .^{[1]:91} As any problem L in NP reduces in polynomial time to G , L reduces in turn to H in polynomial time so this new definition implies the previous one. It does not restrict the class NP-hard to decision problems, for instance it also includes search problems, or optimization problems.

Consequences

- If there is a polynomial algorithm for any NP-hard problem, then there are polynomial algorithms for all problems in NP, and hence $P = NP$;
- If $P \neq NP$, then NP-hard problems cannot be solved in polynomial time, while $P = NP$ does not resolve whether the NP-hard problems can be solved in polynomial time;
- If an optimization problem H has an NP-complete decision version L , then H is NP-hard.

Examples

An example of an NP-hard problem is the decision subset sum problem, which is this: given a set of integers, does any non-empty subset of them add up to zero? That is a decision problem, and happens to be NP-complete. Another example of an NP-hard problem is the optimization problem of finding the least-cost cyclic route through all nodes of a weighted graph. This is commonly known as the traveling salesman problem.

There are decision problems that are NP-hard but not NP-complete, for example the halting problem. This is the problem which asks "given a program and its input, will it run forever?" That is a *yes/no* question, so this is a decision problem. It is easy to prove that the halting problem is *NP-hard* but not *NP-complete*. For example, the Boolean satisfiability problem can be reduced to the halting problem by transforming it to the description of a Turing machine that tries all truth value assignments and when it finds one that satisfies the formula it halts and otherwise it goes into an infinite loop. It is also easy to see that the halting problem is not in *NP* since all problems in NP are decidable in a finite number of operations, while the halting problem, in general, is undecidable. There are also NP-hard problems that are neither NP-complete nor undecidable. For instance, the language of True quantified Boolean formulas is decidable in polynomial space, but not non-deterministic polynomial time (unless $NP = PSPACE$).

NP-naming convention

NP-hard problems do not have to be elements of the complexity class NP, despite having *NP* as the prefix of their class name. The *NP*-naming system has some deeper sense, because the NP family is defined in relation to the class NP and the naming conventions in the Computational Complexity Theory:

NP

Class of computational problems for which a given solution can be verified as a solution in polynomial time by a deterministic Turing machine.

NP-hard

Class of problems which are at least as hard as the hardest problems in NP. Problems in NP-hard do not have to be elements of NP, indeed, they may not even be decidable problems.

NP-complete

Class of problems which contains the hardest problems in NP. Each element of NP-complete has to be an element of NP.

NP-easy

At most as hard as NP, but not necessarily in NP, since they may not be decision problems.

NP-equivalent

Exactly as difficult as the hardest problems in NP, but not necessarily in NP.

Application areas

.

NP-hard problems are often tackled with rules-based languages in areas such as:

- Configuration
- Data mining
- Selection
- Diagnosis
- Process monitoring and control
- Scheduling
- Planning
- Rosters or schedules
- Tutoring systems

- Decision support
- Phylogenetics

References

- [^] ^{**a**} ^{**b**} ^{**c**} Leeuwen, Jan van, ed. (1998). *Handbook of Theoretical Computer Science* (<http://www.worldcat.org/title/algorithms-and-complexity/oclc/247934368/viewport>). Vol. A, Algorithms and complexity. Amsterdam: Elsevier. ISBN 0262720140. OCLC 247934368 (<https://www.worldcat.org/oclc/247934368>).
 - [^] Daniel Pierre Bovet; Pierluigi Crescenzi. *Introduction to the Theory of Complexity*. Prentice Hall. p. 69. ISBN 0-13-915380-2.
 - [^] V. J. Rayward-Smith (1986). *A First Course in Computability*. Blackwell. p. 159. ISBN 0-632-01307-9.
- Michael R. Garey and David S. Johnson (1979). *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W.H. Freeman. ISBN 0-7167-1045-5.

Retrieved from "<http://en.wikipedia.org/w/index.php?title=NP-hard&oldid=642485777>"

Categories: NP-hard problems | Complexity classes

-
- This page was last modified on 14 January 2015, at 18:09.
 - Text is available under the Creative Commons Attribution-ShareAlike License; additional terms may apply. By using this site, you agree to the Terms of Use and Privacy Policy. Wikipedia® is a registered trademark of the Wikimedia Foundation, Inc., a non-profit organization.