**Search problems**

*A problem is a search problem if there's an algorithmic way to verify the answer.*

For example finding a solution to a system of linear equations, finding a solution to a system of linear inequalities,finding a binary solution to the system of boolean equations,finding a non trivial factor of an integer.

That is, whenever we have a solution we can verify it by substituting the value/values.

**NP VS P**

*NP is a set of all search (or decision) problems.*

In other words, all decision problems for which the solutions have proofs verifiable in polynomial time by a deterministic turing machine. The fastest known algorithm for finding the solution may/ may not have polynomial time complexity.

The above mentioned search problems are all NP problems. Some believe that all NP problems have polynomial time solutions which just need to be discovered. However this so far has not been proved.

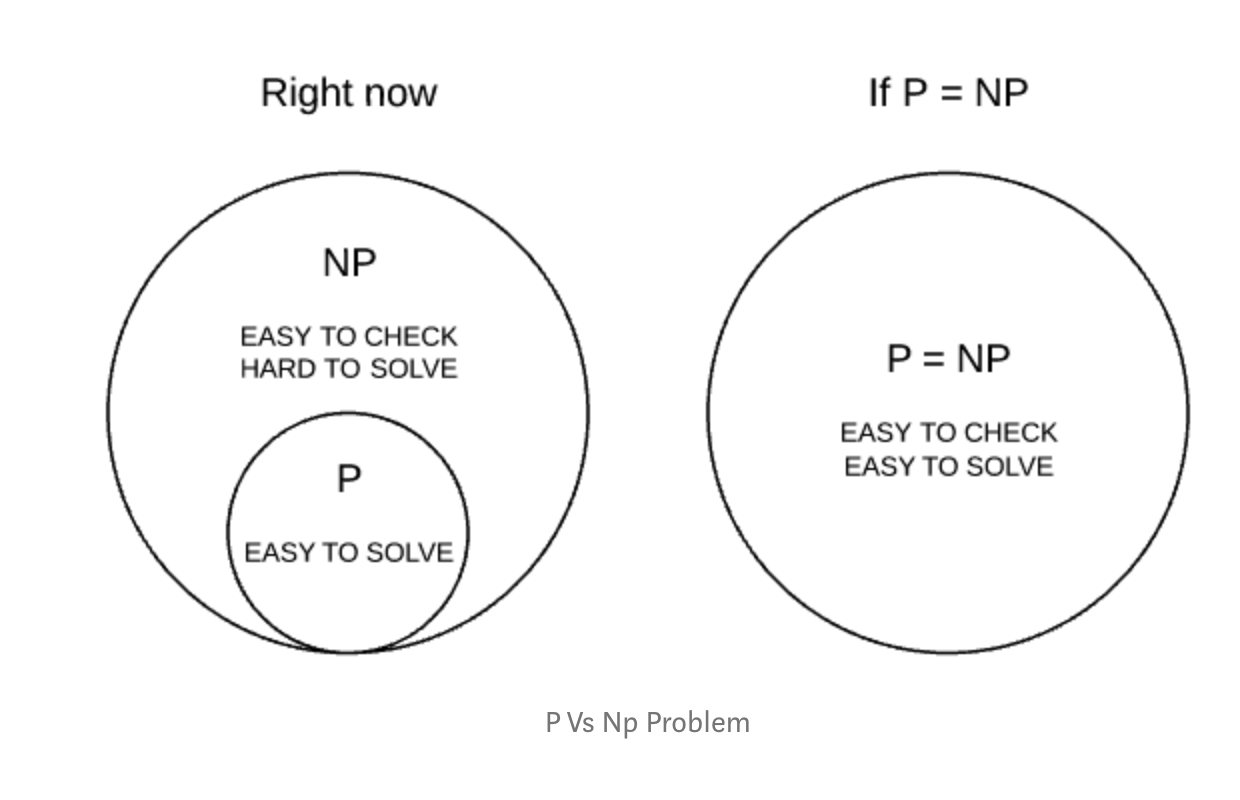
*P is a set of all search problems solvable in polynomial time.*

P is a subset of NP class. For example from the problems mentioned above

* Solving a system of linear equations take n3 time using Gaussian elimination
* Merge sort has a upper bound of n2

Some P problems like solving a system of linear inequalities had no known polynomial time solution for a long time. This may show hope for the equality of P and NP.

**P=NP?**



*Source:*[*https://medium.com/@bilalaamir/p-vs-np-problem-in-a-nutshell-dbf08133bec5*](https://medium.com/@bilalaamir/p-vs-np-problem-in-a-nutshell-dbf08133bec5)

This remains an open question in computer science though now it is widely believed that P≠NP. It is one of the seven Millennium Prize Problems selected by the Clay Mathematics Institute, each of which carries a US$1,000,000 prize for the first correct solution.

If P was indeed equal to P this would indicate that finding the solution of a problem is as easy as verifying a given solution in general, which to me sounds very unintuitive.

**REDUCTION OF PROBLEMS**

Problem X poly-time reduce to problem Y if X can be solved with

* Polynomial number of standard computational steps
* Polynomial number of calls to Y

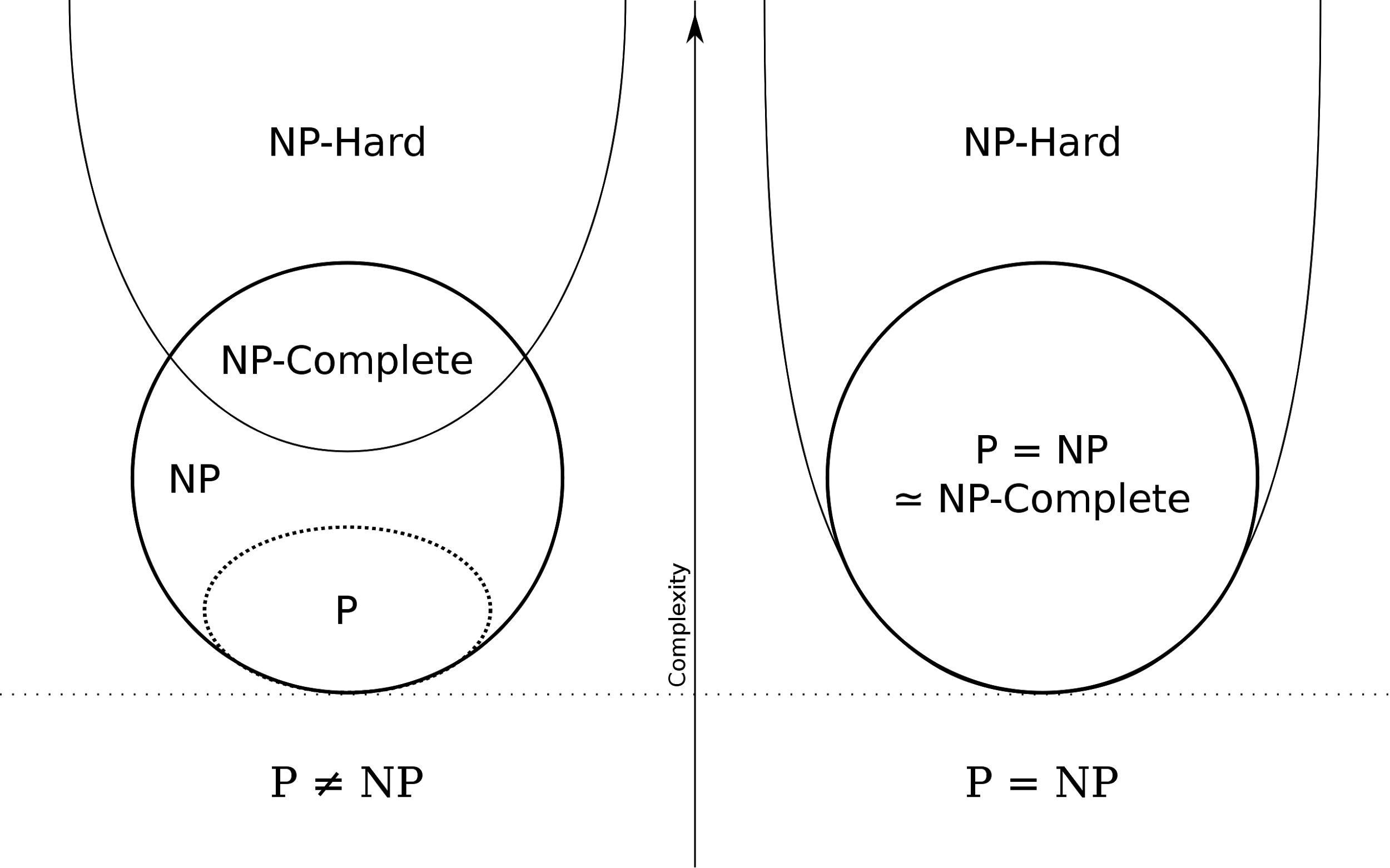
**BOOLEAN SATISFIABILITY PROBLEM**

In computer science, the Boolean satisfiability problem ( SAT) is the problem of determining if there exists a solution to a system of boolean equations. If it does,, the formula is called *satisfiable*. As mentioned above this is a NP but not P problem.

It has been shown that many significant problems such as 0/1 Knapsack, Hamiltonian Path, Traveling Salesman, Vertex cover and many others can be reduced to SAT problems.

This follows that if SAT is intractable then all these are also. In other words these are at least as hard as the SAT problem. Also if one were to find a poly-time algorithm solution for one these problems, then all problems will become P problems.

**NP HARD AND NP COMPLETE PROBLEMS**



*If every problem in NP can be polynomial time reducible to a problem ‘A’ then ‘A’ is called a NP hard problem.*

*If a problem lies in NP and also a NP hard problem then this problem is called NP complete problem.*

SAT is the first problem that was proven to be NP-complete. That is every NP problem can be solved by an indeterministic turing machine which can then be written as an instance of SAT. If we can solve that instance of SAT in poly-time then we solve the original problem too.

In other words a polynomial time solution for SAT exists if and only if P=NP. This is called Cook's theorem which shows that we were to solve one problem the entire class of NP problems will fall.

To conclude most likely a ploy time solution for these problems doesn't exist.

However if someone does prove it..

*"If P were equal to NP, then the world would be a profoundly different place than we usually assume it to be. There would be no special value in "creative leaps", no fundamental gap between solving a problem and recognizing the solution once it's found. Everyone who could appreciate a symphony would be Mozart; Everyone who could follow a set-by-step argument would be Gauss."-* Scott Aaronson,MIT