PhD notes

Matteo Robbiati

Contents

1 Computational Complexity Theory

1.1 Turing Machine

A Turing machine is a mathematical model of computation describing an abstract machine that manipulates symbols on a strip of tape according to a table of rules. Despite the model's simplicity, it is capable of implementing any computer algorithm. A Turing machine is a general example of a central processing unit (CPU) that controls all data manipulation done by a computer, with the canonical machine using sequential memory to store data. More specifically, it is a machine (automaton) capable of enumerating some arbitrary subset of valid strings of an alphabet; these strings are part of a recursively enumerable set. A Turing machine has a tape of infinite length on which it can perform read and write operations.

1.2 Nondeterministic Turing Machine

In theoretical computer science, a nondeterministic Turing machine (NTM) is a theoretical model of computation whose governing rules specify more than one possible action when in some given situations. That is, an NTM's next state is not completely determined by its action and the current symbol it sees, unlike a deterministic Turing machine.

1.3 NP problems

In computational complexity theory, NP (nondeterministic polynomial time) is a complexity class used to classify decision problems. NP is the set of decision problems for which the problem instances, where the answer is "yes", have proofs verifiable in polynomial time by a deterministic Turing machine, or alternatively the set of problems that can be solved in polynomial time by a nondeterministic Turing machine.

1.4 NP-hard problems

In computational complexity theory, NP-hardness (non-deterministic polynomial-time hardness) is the defining property of a class of problems that are informally "at least as hard as the hardest problems in NP". A simple example of an NP-hard problem is the subset sum problem.