WIKIPEDIA

Probabilistic Turing machine

Unsolved problem in computer science:

*Is **P** = **BPP** ?*

(more unsolved problems in computer science)

In computability theory, a **probabilistic Turing machine** is a non-deterministic Turing machine which chooses between the available transitions at each point according to some probability distribution.

In the case of equal probabilities for the transitions, it can be defined as a deterministic Turing machine having an additional "write" instruction where the value of the write is uniformly distributed in the Turing Machine's alphabet (generally, an equal likelihood of writing a '1' or a '0' on to the tape.) Another common reformulation is simply a deterministic Turing machine with an added tape full of random bits called the *random tape*.

As a consequence, a probabilistic Turing machine can (unlike a deterministic Turing Machine) have stochastic results; on a given input and instruction state machine, it may have different run times, or it may not halt at all; further, it may accept an input in one execution and reject the same input in another execution.

Therefore, the notion of acceptance of a string by a probabilistic Turing machine can be defined in different ways. Various polynomial-time randomized complexity classes that result from different definitions of acceptance include RP, co-RP, BPP and ZPP. If the machine is restricted to logarithmic space instead of polynomial time, the analogous RL, co-RL, BPL, and ZPL complexity classes are obtained. By enforcing both restrictions, RLP, co-RLP, BPLP, and ZPLP are yielded.

Probabilistic computation is also critical for the definition of most classes of interactive proof systems, in which the verifier machine depends on randomness to avoid being predicted and tricked by the all-powerful prover machine. For example, the class **IP** equals **PSPACE**, but if randomness is removed from the verifier, we are left with only **NP**, which is not known but widely believed to be a considerably smaller class.

One of the central questions of complexity theory is whether randomness adds power; that is, is there a problem which can be solved in polynomial time by a probabilistic Turing machine but not a deterministic Turing machine? Or can deterministic Turing machines efficiently simulate all probabilistic Turing machines with at most a polynomial slowdown? It is currently widely believed by researchers that the latter is the case, which would imply **P** = **BPP**. The same question for log space instead of polynomial time (does **L** = **BPLP**?) is even more widely believed to be true. On the other hand, the power randomness gives to interactive proof systems, as well as the simple algorithms it creates for difficult problems such as polynomial-time primality testing and log-space graph connectedness testing, suggests that randomness may add power.

A quantum computer is another model of computation that is inherently probabilistic.

See also

- Randomized algorithm

External links

- NIST website on probabilistic Turing machines

Retrieved from "https://en.wikipedia.org/w/index.php?title=Probabilistic_Turing_machine&oldid=827615796"

Foundation, Inc, a non-profit organization.