Turing Machines

Tutorial

Dr S. S. Chandra

Turing machines are more powerful than finite state machines and a problem which is solvable by Turing machines is called Turing decidable. In this tutorial, you will design Turing machines by hand to solve simple problems.

Problem 1

List all the differences between finite state machines and computers we have today with the concept of Turing machines.

Problem 2

Write down the formal description for the Turing machine that decides the language $A = \{w\#w \mid w \in \{0,1\}^*\}$. You may use a transition table or diagram to describe the transition rules for the machine.

Hint: See the example provided by the Turing machine simulator used in the Turing machine laboratory.

Problem 3

Design and formally describe a Turing machine that decides $B = \{0^{2^n} \mid n \geq 0\}$, the language consisting of all strings of 0s whose length is a power of 2. Include

a) a high-level description of its algorithm,
b) a formal description of the Turing machine,
c) a transition/state diagram of the Turing machine,
d) a sample run of the machine on the string 0000 noting its configuration at each step.

Problem 4

Show that the language $C = \{a^i b^j c^k \mid i \times j = k \text{ and } i, j, k \geq 1\}$ cannot be recognisable by a finite state machine by designing a Turing machine that decides it.

Problem 5

Given the differences between finite state machines and Turing machines, and the definition of a Turing machine. In no more than one page (12pt, normal 2.54 cm margins), argue either for or against why a Turing machine can or cannot obtain artificial intelligence (AI). Discuss what needs to be added or removed to ensure that AI can result or can't result. If an AI can result, what restrictions or safe guards do you need in place to protect human-kind from becoming obsolete? If an AI can't result, how will computers and advancements in computing help human-kind in the future?