CS207 Design and Analysis of Algorithms

Sajith Gopalan

Indian Institute of Technology Guwahati

sajith@iitg.ac.in

January 8, 2022

Definition of "algorithm"

- We want to define an algorithm as a mathematical entity like a point, a set, a function
- We define it on "a model of computation"
- Models of computation you might be familiar with:
  - Random Access Macine
  - Turing machine

Random Access Machine

A RAM consists of

- a read-only input tape: a sequence of squares, each of which can hold an integer
- a write-only output tape: ruled into squares which are initially all blank
- a memory: made up of registers capable of holding an integer each; no limit on the number or size of registers; two of the registers are special: accumulator, program counter (the problem is small enough to fit in main memory; integers dealt are small enough to fit in words)
- a program that is not stored in memory, but in a separate read-only store

Random Access Machine Program

- A RAM program is made up of instructions from a RISC-like instruction set
- The precise instruction set does not matter
- Types of instructions: arithmetic, I/O, indirect addressing, branching
- e.g., {LOAD, STORE, ADD, SUB, MULT, DIV, READ, WRITE, GOTO, JZERO, HALT}
- The instructions have the $\langle$ Opcode Operand $\rangle$ format
- The operand could be i, $*i$ or $**i$ (immediate, direct or indirect)
- During the execution of LOAD, STORE, ADD, SUB, MULT, DIV, READ, WRITE program counter is incremented by one
- JUMP: Set the program counter to the operand
- JZERO: if the accumlator is 0, then set the program counter to the operand

Definition 1: Algorithm := RAM program

- An algorithm is a RAM program
- RAM is a mathematical entity, therefore, so is an algorithm
- RAM programs are not easy to write/read
- So we will write our algorithms in English
- with the understanding that they could be easily translated into RAM programs

Turing Machine

A TM $M = (Q, \Sigma, \Gamma, \delta, q, h, \#)$ consists of

- an infinite tape: a sequence of cells, each of which can hold a symbol from a finite alphabet called the tape alphabet Γ
- the input is a string of symbols from a finite alphabet $\Sigma \subset \Gamma$ given on the tape
- a read/write head stationed on some cell of the tape
- a finite set Q of states
- a start state $q \in Q$
- a transition function $\delta : Q \times \Gamma \to Q \times \Gamma \times \{L, S, R\}$
- a halting state $h \in Q$
- a blank symbol $\#$

Turing Machine

A TM M is said to compute a function $f : \Sigma^* \to \Gamma^*$ if and only if for any string $w \in \Sigma^*$, if M starts in state q with w on the input tape (with $\#$s occupying the cells on the either side of w) and the head positioned on the first cell to the left of w, then M will come to halt in state h with $f(w)$ on the tape (with $\#$s occupying the cells on the either side of $f(w)$).

Definition 2: Algorithm := TM

- An algorithm is a TM
- TM is a mathematical entity, therefore, so is an algorithm

RAM $\equiv$ TM

- RAM and TM can simulate each other in polynomial time
- That is, computation that runs in T time on one can be simulated on the other in $T^{O(1)}$ time
- Church's Thesis postulate that TM and equivalent models of computation embody the human computational ability
- Not a mathematical statement, but a philosophical one
- RAM-computable $\equiv$ TM-computable $\equiv$ effectively calculable

Other models of computation

- Many other models have been proposed
- They have all turned out to be less than or equal to TM in computational power
- Strengthens Church's Thesis