ECE 595, Section 10
Numerical Simulations
Lecture 3: Computability

Prof. Peter Bermel

January 11, 2013

Outline

- Overview
- Definitions
- Computing Machines
- Church-Turing Thesis
- Polynomial Time
- Example

Overview from First Lecture

- Study of the complexity of algorithms
- Based on Turing machines
- Often, one compares algorithms for best scaling in large problems

Alan Turing (from University of Calgary Centenary event)

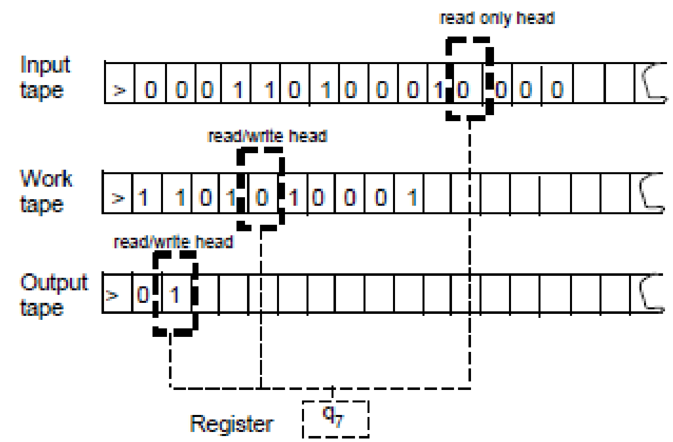
Definitions

- Algorithm – set of rules for computing a function mapping input data to output
- Specific realizations of computers:
  - Turing machines
  - Lambda calculus
  - μ-recursive functions
  - Register machines
- Running time – number of basic operations required
- Efficient computability

Turing Machine

- Performs the following elementary operations:
  - Read a bit of input and byte of data from 'work'
  - Write a bit to 'work'
  - Choose next state *Q.* Either:
    - Stop and output a 0 or 1
    - Choose a new rule to apply next

$$\delta : Q \times \Gamma^k \to Q \times \Gamma^{k-1} \times \{\mathsf{L}, \mathsf{S}, \mathsf{R}\}^k$$



Turing Machine: Behavior in Each State

- q_{start}: write start symbol, change state to q_{copy}
- q_{copy}: copy non-blank symbol, otherwise go to q_{right}
- q_{right}: go back to start, then q_{test}
- q_{test}: if at start and 'work' is blank, write 1; else, write 0
- q_{halt}: finished

Time Constructible Functions

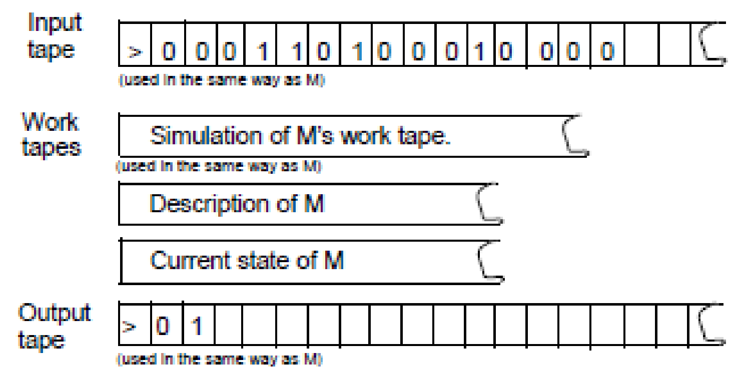
- A function T: N → N is time constructible iff
  - $T(n) \geq n$
  - TM M computes $x \rightarrow T(x)$ in time $T(n)$
- Examples: n, n log n, 2^n

Turing Machine: Key Properties

- If f is computable in T(n)….
  - with alphabet Γ, it is computable in 4 log Γ T(n) with TM
  - with k tapes, it's computable in $5kT(n)^2$ with TM
  - with bidirectional tape, computable in 4T(n) with TM

Universal Turing Machines

- A universal Turing machine can simulate every other TM
- Efficient process: takes $CN \log N$ time to simulate a process taking N steps



Other Computing Machines

- Register machines
  - Post-Turing machine: adds jump and erase functions
  - Many other variations, getting closer to assembly language
- Lambda calculus – basis for Scheme and LISP
- μ-recursive functions – partial functions mapping $N \rightarrow N$
- Which one should we use, in which situations?

Church-Turing Thesis

- Every physically realizable computing machine can be simulated by a Turing machine, or any other type of machine
- This thesis has not been rigorously proven
- Not applicable for non-deterministic systems
- Quantum computers have caused some to re-evaluate CT thesis

Uncomputability

- It may seem obvious everything is computable with enough time, but…
- There's always an uncomputable function UC
- Another example: the halting problem
  - Computes whether another function will finish
  - *Cannot be computed in general*
- Also: standard maximum productivity function

Big-Oh Notation

- Take two functions f,g mapping N→N, there exists a constant c such that $f(n) \le c \cdot g(n)$
  - Sometimes write $f(n) = O(g(n))$
- Examples
  - If $f(n)=100n^2+24n+2\log n$ and $g(n)=n^2$, then $f=O(g)$
  - If $f(n)=2^n$ and $g(n)=n^c$ for every c in N, then $g=O(f)$

Polynomial Complexity

- Meant to capture decision problems that are feasible
- Might think of efficient computations as being O(N) or O(N^2)
- Often symbolized by **P**
- Formally, $\mathrm{P} = \cup_{c \geq 1} \mathbf{DTIME}(n^c)$, where **DTIME**(T) is computable in cT(n) time
- Strong form of CT thesis says all simulations can be done with **P** overhead

Other Classes

- Turing machines with randomness – class BPP
- Quantum computers – class BQP

Examples

- CONNECTED – the set of all connected graphs
- TRIANGLEFREE – the set of all graphs without a triangle
- BIPARTITE – the set of all bipartite (distinct) graphs
- TREE – the set of all trees

Next Class

- Discussion of computational complexity
- Please read Chapter 2 of "Computational Complexity: A Modern Approach" by Arora & Barak