Discrete Mathematical Structures
CS 3233 Lecture 12

Prof. William Winsborough
October 16, 2007

Business

- Read sections 3.1 and 3.2 for Thursday
- Recall: Homework 7 due Thursday 10/18
  - Section 2.3: 10, 14, 16, 38
  - Section 2.4: 4, 8

Sequences

- Definition
  - A *sequence* is a function from $\mathbb{N}$ or $\mathbb{Z}^+$ to a given set S
    - We use a_n to denote the image of n
    - We use $\{a_n\}$ to denote the whole sequence
      - Less formally, we sometimes denote it by $\{a_0, a_1, a_2, a_3, \ldots\}$
  - If the function is onto we say $\{a_n\}$ enumerates S

Countability

- Definition
  - A set A is *countable* if it is finite or it is equinumerous to Z^+
  - Otherwise, A is *uncountable*
- Theorem: Q^+, the set of positive rationals, is countable

Countability and Enumeration

- Theorem: S is countable if and only if there exists a sequence that enumerates S
- Proof
  - Only if: If there is a bijection between N and S, it is a sequence that enumerates S
  - If: Given a sequence that enumerates S, either S is finite or dropping repeated values from the sequence yields a bijection between N and S

Uncountability of the Reals

- Theorem
  - R, the set of real numbers, is uncountable
- Proof
  - Uses Georg Cantor's *diagonalization argument*
  - Outline
    - Assume for contradiction that there is a one-to-one correspondence, f, between N and the real interval [0,1]
    - Use f to construct a real in [0,1] that has no preimage under f
    - Idea: for each decimal place, n, in the representation of the constructed value, choose a decimal different from the n^{th} place of f(n)
    - The fact that the constructed value differs from each value assumed by f shows that f is not onto, giving the desired contradiction

Countability Exercise 1

- Theorem: If A is an uncountable set and B is a countable set, A – B is uncountable
- Proof
  - Suppose for contradiction that A – B is countable
  - This means that there is a sequence that enumerates all elements of A – B
  - We can now construct a sequence that enumerates A
    - It alternates between the sequence that enumerates A – B and the sequence that enumerates B
    - This contradicts the assumption that A is uncountable
  - It follows that the assumption A – B is countable is false, hence A – B is uncountable

Countability Exercise 2

- Theorem: If A is an uncountable set and A is a subset of B, then B is uncountable
- Proof:
  - Suppose B is countable
  - This means there is a sequence that enumerates B
  - A sequence that enumerates A can be constructed by dropping the elements of B – A, yielding the desired contradiction

Algorithms

- Definition
  - An *algorithm* is a finite collection of precise instructions for performing a computation to solve a problem
  - Input and output values are elements of specified sets
- Desirable characteristics:
  - Definiteness: Steps are precisely defined
    - To be really precise, must use a formal *computational model,* such as a Turing machine or the lambda calculus
  - Effectiveness: It must be possible to perform each step using a bounded amount of time and storage space
    - Bounded means there is an amount of time that is always sufficient
  - Correctness, Termination (Finiteness), Generality

Examples

- Searching: Determine whether a given value is contained in an input sequence of integers

Linear Search

proc linear search(x:int; $a_1,a_2,\ldots,a_n$: distinct ints)
i := 1
while ($i \le n$ and $x \ne a_i$)
  i := i + 1
if $i \le n$ **then** location := i **else** location := 0

- How many comparisons are performed?

Binary Search

proc binary search(x: int; $a_1, a_2, \ldots, a_n$: <u>increasing</u> ints)
i:=1 {left end of search interval}
j:=n {right end of search interval}
while i<j **begin**
  m:= $\lfloor (i+j)/2 \rfloor$
  if $x > a_m$ **then** i:=m+1 **else** j:= m
  end
if $x=a_i$ **then** location := i **else** location := 0

- How many comparisons are performed?
  - This will be studied in Section 3.3

Bubble Sort

proc bubble sort($a_1,a_2,\ldots,a_n$)
for i := 1 to n-1
  for j := 1 to n-i
  if $a_j > a_{j+1}$ **then** interchange a_j and a_{j+1}
{$a_1,a_2,\ldots,a_n$ is in increasing order}

- How many comparisons are performed?

Insertion Sort

proc insertion sort($a_1,a_2,\ldots,a_n$: reals with $n \geq 2$)
for j := 2 **to** n
begin
  i := 1
  while $a_j > a_i$
    i := i+1
  m := a_j
  for k := 0 **to** j – i – 1
    $a_{j-k} := a_{j-k-1}$
  a_i := m
end {$a_1,a_2,\ldots,a_n$ is in increasing order}

Growth of Functions

- Big-O Notation
- Def: Let f, g : Z → R. We say f(x) is O(g(x)) if there are constants C and k such that $|f(x)| \leq C|g(x)|$ for all x > k
  - We say “f(x) is big-oh of g(x)”
- When discussing positive-valued functions, we can drop the |·|

Example

- Let $f(x) = x^2 + 2x + 1$ and $g(x) = x^2$
- f(x) is O(g(x))
  - k = 1 and C = 4 (witnesses)
  - k = 2 and C = 3
- g(x) is O(f(x))
  - k = 1 and C = 1

Example

- Let $f(x) = 7x^2$ and $h(x) = x^3$
- $7x^2$ is $O(x^3)$
  - k = 1 and C = 7
- x^3 is not $O(7x^2)$
  - There is no C such that $x^3 \leq C(7x^2)$ for “large” x

Degree of Polynomial is What Matters

- Theorem:
  Let $f(x) = a_nx^n + a_{n-1}x^{n-1} + \ldots + a_1x + a_0$.
  Then f(x) is $O(x^n)$

More Examples

- Sum of first n positive integers
  $1+2+\ldots+n \le \underbrace{n+n+\ldots+n}_{n} = n^2$
  So $\sum_{i=1}^{n} i$ is $O(n^2)$
- It follows that the complexity of both bubble sort and insertion sort is $O(n^2)$

Still More Examples

- $f(n) = n!$ is the product of the first n positive integers
  $1 \cdot 2 \cdot \ldots \cdot n \le \underbrace{n \cdot n \cdot \ldots \cdot n}_{n} = n^n$

  So $n!$ is $O(n^n)$

Big-O and Sums

- Def: If f_1, f_2 are real-valued functions, (f_1+f_2) is the function such that that $(f_1+f_2)(x) = f_1(x)+f_2(x)$ for all x
- Theorem:
  If $f_1(x)$ is $O(g_1(x))$ and $f_2(x)$ is $O(g_2(x))$, then $(f_1+f_2)(x)$ is $O(\max(|g_1(x)|, |g_2(x)|)$
- Corollary:
  If $f_1(x)$ and $f_2(x)$ are each $O(g(x))$, then $(f_1+f_2)(x)$ is $O(g(x))$

Big-O and Products

- Def: If f_1, f_2 are real-valued functions, (f_1f_2) is the function such that that $(f_1f_2)(x) = f_1(x) \cdot f_2(x)$ for all x
- Theorem:
  If $f_1(x)$ is $O(g_1(x))$ and $f_2(x)$ is $O(g_2(x))$, then $(f_1f_2)(x)$ is $O(g_1(x)\ g_2(x))$