Discrete Mathematical Structures
CS 3233 Lecture Nine

Prof. William Winsborough
February 19, 2008

Business

- **Homework 5**, due Tuesday 2/26
  - 2.4: 2, 32, 36
  - 3.1: 2, 4, 12, 18, 24
- Read Sections 2.4 and 3.1
- Midterm 1 is Thursday 2/28
  - Mock exam will be posted on Course Web Page
- Tuesday 2/26 will be devoted to review
  - Student questions will determine the agenda
  - Come prepared
- Questions???
  - Homework 4?

Countability

- *Recall:* Sets A and B are *equinumerous* (meaning they have the same cardinality) iff there is a one-to-one correspondence between them
- Definition
  - A set A is *countable* if it is finite or it is equinumerous to $\mathbb{N}$
  - Otherwise, A is *uncountable*

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 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
- Theorem: Q^+, the set of positive rationals, is countable

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 $x \neq a_i$)
  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