Discrete Mathematics

Administration

- Class Web Site

  http://cs.nyu.edu/courses/summer16/CSCI-GA.2340-001/
- Mailing List

  Subscribe at see website

  Messages to: see website
- TA/Office Hours, etc: Jianbo Sun
- Homework

Disproofs, Counterexamples and Algebraic Proofs

- Is is true that $(A - B) \cup (B - C) = A - C$? (No via counterexample)
- Show that $(A \cup B) - C = (A - C) \cup (B - C)$ (Can do with an algebraic proof, slightly different)

Russell's Paradox

- Set of all integers, set of all abstract ideas
- Consider S = {A, A is a set and $A \notin A$}
- Is S an element of S?
- Barber puzzle: a male barber shaves all those men who do not shave themselves. Does the barber shave himself?
- Consider S = {$A \subseteq U$, $A \notin A$}. Is $S \in S$?
- Godel: No way to rigorously prove that mathematics is free of contradictions. ("This statement is not provable" is true but not provable) (consistency of an axiomatic system is not provable within that system)

Halting Problem

- There is no computer algorithm that will accept any algorithm X and data set D as input and then will output “halts” or “loops forever” to indicate whether X terminates in a finite number of steps when X is run with data set D.
- Proof is by contradiction

Generic Functions

- A function f: X → Y is a relationship between elements of X to elements of Y, when each element from X is related to a unique element from Y
- X is called domain of f, range of f is a subset of Y so that for each element y of this subset there exists an element x from X such that y = f(x)
- Sample functions:
  - f : R → R, $f(x) = x^2$
  - f : Z → Z, f(x) = x + 1
  - f : Q → Z, f(x) = 2

Generic Functions

- Arrow diagrams for functions
- Non-functions
- Equality of functions:
  - $f(x) = |x|$ and $g(x) = \text{sqrt}(x^2)$
- Identity function
- Logarithmic function

One-to-One Functions

- Function $f : X \rightarrow Y$ is called one-to-one (injective) when for all elements x_1 and x_2 from X if $f(x_1) = f(x_2)$, then $x_1 = x_2$
- Determine whether the following functions are one-to-one:
  - $f : R \rightarrow R$, $f(x) = 4x - 1$
  - $g : Z \rightarrow Z$, $g(n) = n^2$
- Hash functions

Onto Functions

- Function $f : X \rightarrow Y$ is called onto (surjective) when given any element y from Y, there exists x in X so that $f(x) = y$
- Determine whether the following functions are onto:
  - $f : R \rightarrow R$, $f(x) = 4x - 1$
  - $f : Z \rightarrow Z$, $g(n) = 4n - 1$
- Bijection is one-to-one and onto
- Reversing strings function is bijective

Inverse Functions

- If $f : X \rightarrow Y$ is a bijective function, then it is possible to define an inverse function $f^{-1}: Y \rightarrow X$ so that $f^{-1}(y) = x$ whenever $f(x) = y$
- Find an inverse for the following functions:
  - String-reverse function
  - $f : R \rightarrow R$, $f(x) = 4x – 1$
- Inverse function of a bijective function is a bijective function itself

Composition of Functions

- Let f : X → Y and g : Y → Z, let range of f be a subset of the domain of g. The we can define a composition of g o f : X → Z
- Let f,g : Z → Z, f(n) = n + 1, g(n) = n^2. Find f o g and g o f. Are they equal?
- Composition with identity function
- Composition with an inverse function
- Composition of two one-to-one functions is one-to-one
- Composition of two onto functions is onto

Pigeonhole Principle

- If n pigeons fly into m pigeonholes and n > m, then at least one hole must contain two or more pigeons
- A function from one finite set to a smaller finite set cannot be one-to-one
- In a group of 13 people must there be at least two who have birthday in the same month?
- A drawer contains 10 black and 10 white socks. How many socks need to be picked to ensure that a pair is found?
- Let A = {1, 2, 3, 4, 5, 6, 7, 8}. If 5 integers are selected must at least one pair have sum of 9?

Pigeonhole Principle

- Generalized Pigeonhole Principle: For any function f : X → Y acting on finite sets, if n(X) > k * N(Y), then there exists some y from Y so that there are at least k + 1 distinct x's so that f(x) = y
- "If n pigeons fly into m pigeonholes, and, for some positive k, m >k*m, then at least one pigeonhole contains k+1 or more pigeons"
- In a group of 85 people at least 4 must have the same last initial.
- There are 42 students who are to share 12 computers. Each student uses exactly 1 computer and no computer is used by more than 6 students. Show that at least 5 computers are used by 3 or more students.

Cardinality

- Cardinality refers to the size of the set
- Finite and infinite sets
- Two sets have the same cardinality when there is bijective function associating them
- Cardinality is is reflexive, symmetric and transitive
- Countable sets: set of all integers, set of even numbers, positive rationals (Cantor diagonalization)
- Set of real numbers between 0 and 1 has same cardinality as set of all reals
- Computability of functions

Algorithms

- Algorithm is step-by-step method for performing some action
- Cost of statements execution
  - Simple statements
  - Conditional statements
  - Iterative statements

Division Algorithm

- Input: integers **a** and **d**
- Output: quotient **q** and remainder **r**
- Body:

```
r = a; q = 0;
while (r >= d)
    r = r – d;
    q = q + 1;
end while
```

Greatest Common Divisor

- The greatest common divisor of two integers a and b is another integer d with the following two properties:
  - d | a and d | b
  - if c | a and c | b, then $c \leq d$
- Lemma 1: gcd(r, 0) = r
- Lemma 2: if a = b * q + r, then gcd(a, b) = gcd(b, r)

Euclidean Algorithm

- Input: integers **a** and **b**
- Output: greatest common divisor **gcd**
- Body:

```
r = b;
while (b > 0)
    r = a mod b;
    a = b;
    b = r;
end while
gcd = a;
```

Exercise

- Least common multiple: lcm
- Prove that for all positive integers a and b, gcd(a, b) = lcm(a, b) iff a = b

Correctness of Algorithms

- Assertions
  - Pre-condition is a predicate describing initial state before an algorithm is executed
  - Post-condition is a predicate describing final state after an algorithm is executed
- Loop guard
- Loop is defined as correct with respect to its pre- and post- conditions, if whenever the algorithm variables satisfy the pre-conditions and the loop is executed, then the algorithm satisfies the post-conditions as well

Loop Invariant Theorem

- Let a while loop with guard G be given together with its pre- and post- conditions. Let predicate I(n) describing loop invariant be given. If the following 4 properties hold, then the loop is correct:
  - Basis Property: I(0) is true before the first iteration of the loop
  - Inductive Property: If G and I(k) is true, then I(k + 1) is true
  - Eventual Falsity of the Guard: After finite number of iterations, G becomes false
  - Correctness of the Post-condition: If N is the least number of iterations after which G becomes false and I(N) is true, then post-conditions are true as well

Correctness of Some Algorithms

- Product Algorithm:

  pre-conditions: $m \geq 0$, $i = 0$, product = 0

  while ($i < m$) {

  product += x;

  i++;

  }

  post-condition: product = m * x

Correctness of Some Algorithms

- Division Algorithm

  pre-conditions: $a \geq 0$, $d > 0$, $r = a$, $q = 0$

  while ($r \geq d$) {

  r -= d;

  q++;

  }

  post-conditions: $a = q * d + r$, $0 \leq r < d$

Correctness of Some Algorithms

- Euclidean Algorithm

  pre-conditions: $a > b \geq 0$, $r = b$

  while ($b > 0$) {

  $r = a \bmod b$;

  $a = b$;

  $b = r$;

  }

  post-condition: $a = \gcd(a, b)$

Matrices

- Sum of two matrices A and B (of size mxn) – Ex.
- Product of mxk matrix A and kxn matrix B is a mxn matrix C – Examples.
- Body:

```
for i := 1 to m
  for i := 1 to n
      c_ij := 0
      for q := 1 to k
         c_ij := c_ij + a_iq*b_qj
end
Return C
```

Sequences

- Sequence is a set of (usually infinite number of) ordered elements: a_1, a_2, …, a_n, …
- Each individual element a_k is called a term, where k is called an index
- Sequences can be computed using an explicit formula: $a_k = k * (k + 1)$ for $k > 1$
- Alternate sign sequences
- Finding an explicit formula given initial terms of the sequence: 1, -1/4, 1/9, -1/16, 1/25, -1/36, …
- Sequence is (most often) represented in a computer program as a single-dimensional array

Sequence Operations

- Summation: Σ, expanded form, limits (lower, upper) of summation, dummy index
- Change of index inside summation
- Product: Π , expanded form, limits (lower, upper) of product, dummy index
- Factorial: n!, n! = n * (n – 1)!

Sequences

- Geometric sequence:
  $a, ar, ar^2, ar^3, \ldots, ar^n$
- Arithmetic sequence:
  $a, a+d, a+2d, \ldots, a+nd$
- Sum of geometric sequence:
  $\sum_{0->n} ar^k$
- Sum of arithmetic sequence:
  $\sum_{0->n} a+kd$

Review Mathematical Induction

- Principle of Mathematical Induction:

  Let P(n) be a predicate that is defined for integers n and let a be some integer. If the following two premises are true:

  P(a) is a true

  $\forall k \geq a, P(k) \rightarrow P(k + 1)$

  then the following conclusion is true as well

  P(n) is true for all $n \geq a$

Applications of Mathematical Induction

- Show that $1 + 2 + \ldots + n = n * (n + 1) / 2$
  (Prove on board)
- Sum of geometric series:

  $r^0 + r^1 + \ldots + r^n = (r^{n+1} - 1) / (r - 1)$
  (Prove on board)

Examples that Can be Proved with Mathematical Induction

- Show that $2^{2n} - 1$ is divisible by 3 (in book)
- Show (on board) that for $n > 2$: $2n + 1 < 2^n$
- Show that $x^n - y^n$ is divisible by $x - y$
- Show that $n^3 - n$ is divisible by 6 (similar to book problem)

Strong Mathematical Induction

- Utilization of predicates P(a), P(a + 1), …, P(n) to show P(n + 1).
- Variation of normal M.I., but basis may contain several proofs and in assumption, truth assumed for all values through from base to k.
- Examples:
- Any integer greater than 1 is divisible by a prime
- Existence and Uniqueness of binary integer representation (Read in book)

Well-Ordering Principle

- Well-ordering principle for integers: a set of integers that are bounded from below (all elements are greater than a fixed integer) contains a least element
- Example:
- Existence of quotient-remainder representation of an integer n against integer d