BROWN
Computer Science

CS1570: Design and Analysis of Algorithms

Lecture 0b: Mathematical Preliminaries Review

Lorenzo De Stefani

Fall 2020

Outline

Review of math concepts

- Sets
- Sequences and Tuples
- Functions and Relations
- Graphs
- Boolean Logic
- Random variables and expectation

Sets

A set is a group of objects, order does not matter

- The objects are called elements or members
- Examples:
  - {1, 3, 5}, {1, 3, 5, …}, or $\{x \mid x \in Z \text{ and } x \bmod 2 \neq 0\}$
- You should know these operators/concepts
  - Subset ($A \subset B$ or $A \subseteq B$)
  - Cardinality: Number elements in set ($|A|$)
  - Intersection ($\cap$) and Union ($\cup$), Complement $\bar{A}$
  - Venn Diagrams: can be used to visualize sets

Sets II

Power Set: All possible subsets of a set

– If A = {0, 1} then what is P(A)?
– In general, what is the cardinality of P(B)?

Sets II

Power Set: Set of all possible subsets of a set

- If A = {0, 1} then what is P(A)?
  - $P(A) = \{\emptyset, \{0\}, \{1\}, \{0,1\}\}$
- In general, what is the cardinality of P(B)?
  - Number of the possible binary strings with |B| bits

    $|P(B)| = 2^{|B|}$.

Sequences and Tuples

- A sequence is a list of objects, <u>order matters</u>
  - Example: (1, 3, 5) or (5, 3, 1)
- In this course we will use term "tuple" instead
  - (1, 3, 5) is a 3-tuple and a *k*-tuple has *k* elements

Sequences and Tuples II

Cartesian product (x) is an operation on sets but yields a set of tuples

- Example: if $A = \{1, 2\}$ and $B = \{x, y, z\}$
  - $A \times B = \{(1,x), (1,y), (1,z), (2,x), (2,y), (2,z)\}$
- If we have k sets $A_1, A_2, \ldots, A_k$, we can take the Cartesian product $A_1 \times A_2 \ldots \times A_k$ which is the set of all k-tuples $(a_1, a_2, \ldots, a_k)$ where $a_i \in A_i$
- We can take Cartesian product of a set with itself
  - A^k represents $A \times A \times A \ldots \times A$ where there are k A's.
- The set Z^2 represents $Z \times Z$ all pairs of integers, which can be written as $\{(a,b) \mid a \in Z \text{ and } b \in Z\}$

Functions and Relations

- A function maps an input to a (single) output
  - $f(a) = b$, f maps a to b
- The set of possible inputs is the domain and the set of possible outputs is the range
  - $f : D \to R$
  - Example 1: for the *abs* function, if D = Z, what is R?
  - Example 2: *sum* function
    - $f_{sum} : Z \times Z \to Z$
- Functions can be described using tables
  - Example: Describe $f(x) = 2x$ for $D=\{1,2,3,4\}$

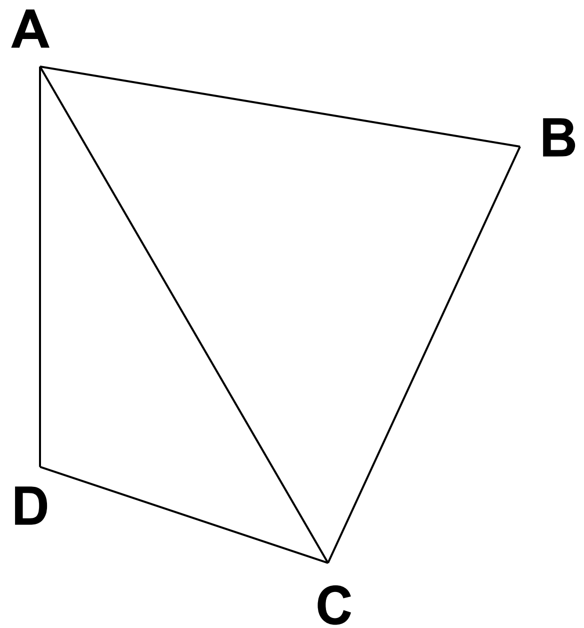
Relations and predicates

- A "predicate" is a function with range {True, False}
  - Example: even(4) = True
- A (k-ary) relation is a predicate whose domain is a set of k-tuples A x A x A … x A
  - If k =2 then binary relation (e.g., =, <, ...)
- Relations may have 3 key properties:
  - reflexive, symmetric, transitive
  - A binary relation is an equivalence relation if it has all 3
  - Try =, <, friend

Graphs

A graph is a set of vertices V and edges E

– G= (V,E) and can use this to describe a graph



V = {A, B, C, D}

E = {(A,B), (A,C), (C,D), (A,D), (B,C)}

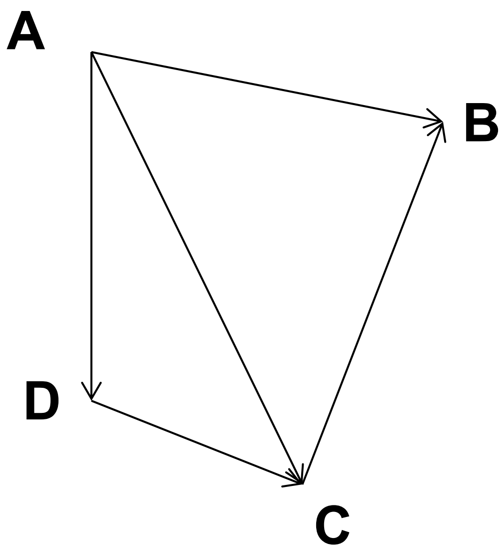
Graphs

Definitions:

- The degree of a vertex is the number of edges touching it
- A path is a sequence of nodes connected by edges
- A simple path does not repeat nodes
- A path is a cycle if it starts and ends at same node
- A simple cycle repeats only first and last node
- A graph is a tree if it is connected and has no simple cycles

Directed Graphs

A directed graph G= (V,E) is a set of vertices V and directed edges E



V = {A, B, C, D}
E = {(A,B), (A,C), (D,C), (A,D), (C,B)}
... edges are denotes as tuples: (A,B) denotes an edge directed from A to B

Directed Graphs

Definitions:

- The in-degree of a vertex is the number of edges entering it
- The out-degree of a vertex is the number of edges exiting it
- A directed path is an ordered sequence of nodes connected by directed edges
- A simple directed path does not repeat nodes
- A path is a directed cycle if it starts and ends at same node
- A simple cycle repeats only first and last node
- A graph is a tree if it is connected and has no simple cycles
- A directed graph with no directed cycles is said to be a Directed Acyclic Graph (DAG)

Boolean Logic

- Boolean logic is a mathematical system built around True (or 1) and False (or 0)
- Below are the boolean operators, which can be defined by a truth table
  - $\wedge$ (and/conjunction) $1 \wedge 1 = 1$; else 0
  - $\vee$ (inclusive or/disjunctions) $0 \vee 0 = 0$; else 1
  - $\neg$ (not) $\neg 1 = 0$ and $\neg 0 = 1$
  - $\rightarrow$ (implication) $1 \rightarrow 0 = 0$; else 1
  - $\leftrightarrow$ (equality) $1 \leftrightarrow 1 = 1$; $0 \leftrightarrow 0 = 1$
- Can prove equality using truth tables
  - DeMorgan's law and Distributive law

Probability spaces

A Probability Space has three components:

- A Sample Space Ω, which is the set of all possible outcomes of the random process being observed
- A family of sets F representing the the allowable events, where each set in F is a subset of Ω
  - Elements of F also referred as "Events"
  - Elements of Ω referred as "Elementary events" or "Samples"
- A probability function Pr: F→ R which satisfies the properties:
  - For any $E \in F$, $O \leq \Pr(E) \leq 1$
  - $\Pr(\Omega) = 1$
  - For any finite or countably infinite sequence of pairwise disjoint events $E_1, E_2, E_3, \ldots$

$$\Pr(\cup E_i) = \Sigma \Pr(E_i)$$

Fundamental properties of probability functions

- For any two events $E_1, E_2 \in \mathrm{F}, \Pr(\mathrm{E}_1 \cup E_2) = \Pr(\mathrm{E}_1) + \Pr(\mathrm{E}_2) - \Pr(\mathrm{E}_1 \cap E_2)$
- For any finite or countably finite sequence of events $E_1, E_2, E_3, \ldots$ we have
$$\Pr(\cup E_i) \leq \sum_i \Pr(E_i)$$
- Two events $E_1, E_2 \in \mathrm{F}$ are independent if and only if
$$\Pr(\mathrm{E}_1 \cap E_2) = \Pr(\mathrm{E}_1)\Pr(\mathrm{E}_2)$$
  ... independence generalizes to multiple events
- The conditional probability that event E_1 occurs given that event E_2 occurs is
$$\Pr(\mathrm{E}_1 | E_2) = \Pr(\mathrm{E}_1 \cap E_2)/\Pr(\mathrm{E}_2)$$

Fundamental properties of probability functions

- Law of Total Probability: Let $E_1, E_2, E_3, \ldots$ be mutually disjoint events in the sample space Ω, and let $\cup E_i = \Omega$. That is the subsets $E_1, E_2, E_3, \ldots$ partition Ω, then for any $B \subseteq \Omega$:

$$\Pr(B) = \Sigma \Pr(B \cap E_i) = \Sigma \Pr(B|E_i) \Pr(E_i)$$

- Bayes' Law:

$$\Pr(E_i|B) = \frac{\Pr(E_i \cap B)}{\Pr(B)} = \frac{\Pr(B|E_i)\Pr(E_i)}{\Sigma \Pr\left(B\middle|E_j\right) \Pr(E_j)}$$

Random Variables

- Sample space Ω: set of values which represent outcomes of an experiment
- A random variable X on a sample space Ω is a real-valued function on Ω, X: Ω→*R*
- A discrete random variable, is a random variable that can only assume a finite or countably infinite number of values.
- Given a discrete random variable X and a real value a: the event "$X = a$" represents the sub set of Ω given by $\{s \in \Omega : X(s) = a\}$

$$\Pr(X = a) = \Sigma_{s \in \Omega : X(s) = a} \Pr(s)$$

Independence

Two random variables X and Y are independent if and only if

$$\Pr\big((X = x) \cap (Y = y)\big) = \Pr(X = x) \Pr(Y = y)$$

for all values x and y. The definition extends to multiple random variables.

Expectation

The expectation of a discrete random variable X, denoted as $E[X]$ is given by

$$E[X] = \Sigma i \, Pr(X = i)$$

where the summation is over the values i in the range of X.

- $E[X]$ is a weighted sum over all possible values weighted according to their probability
- The expectation is finite if $\Sigma|i| \, Pr(X = i)$ converges to a finite value, otherwise it is unbounded
- For any pair of random variables X_1, X_2 and constants a, b we have, by linearity of expectation

$$E[aX_1 + bX_2] = aE[X_1] + bE[X_2]$$

Randomized algorithms

- Knowledge of basic notions of probability theory will be useful to design and analyze Randomized Algorithms
- An algorithm is said to be “randomized” if it some of its decisions or operations are influenced by the outcome of a random experiment
  - coin flip, extraction of a value, sampling, ...
- Two main types of random algorithms:
  - Las Vegas:
    - Always return correct solution
    - Execution time is a random variable!
    - We generally care about the Expected running time
    - E.g., Quicksort, random selection
  - Montecarlo:
    - Has a certain failure probability → may return a wrong answer
    - Generally there is a trade-off between the quality of the returned answer and its probability
    - The worse the quality of the output, the lower its probability
    - Execution time is always the same!
    - Very used in statistical learning and approximation algorithms