CS 570 Graph Theory

Lecture 0: The Foundations

Lecturer: Uğur Doğrusöz CS Dept., Bilkent University

1 The Foundations: Discrete Mathematics

1.1 Principles of Counting

The study of discrete and combinatorial mathematics begins with two basic principles of counting:

- *The Rule of Sum:* If a task can be performed in m ways, while a second task can be performed in n ways, and the two tasks cannot be performed simultaneously, then performing either task can be accomplished in any one of $m+n$ ways.
- *The Rule of Product:* If a procedure can be broken down into first and second stages, and if there are m possible outcomes for the first stage and if, for each of these outcomes, there are n possible outcomes for the second stage, then the total procedure can be carried out, in the designated order, in mn ways.

Permutations: In general, if there are n distinct objects, denoted $a_1, a_2, \ldots, a_n$, and r is an integer, with $1 \leq r \leq n$, then by the rule of product, the number of permutations of size r for the n objects is

$$n \cdot (n-1) \cdot (n-2) \cdot \ldots \cdot (n-r+1) =$$

$$(n)(n-1)(n-2)\ldots(n-r+1) \cdot \frac{(n-r)(n-r-1)\ldots(3)(2)(1)}{(n-r)(n-r-1)\ldots(3)(2)(1)} = \frac{n!}{(n-r)!}$$

Combinations: In general, if we start with n distinct objects, each *selection* or *combination*, of r of these objects, with no reference to order, corresponds to $r!$ permutations of size r from the n objects. Thus the number of combinations of size r from a collection of size n, denoted $C(n,r)$, where $0 \leq r \leq n$, satisfies $r! \cdot C(n,r) = P(n,r)$ and

$$C(n,r) = \frac{P(n,r)}{r!} = \frac{n!}{r!(n-r)!}, \quad 0 \leq r \leq n$$

In addition to $C(n,r)$ the symbol $\binom{n}{r}$ is frequently used, both reading "n choose r". Note that $C(n,0) = 1$, for all $n \geq 0$.

1.2 Logic

The rules of logic give precise meaning to mathematical statements. These rules are used to distinguish between valid and invalid mathematical arguments.

A **proposition** is a statement that is either true (T - *tautology*) or false (F - *contradiction*).

Compound propositions can be formed using logical operators such as $\vee$, $\wedge$, $\oplus$, $\rightarrow$. Equivalence of two propositions is expressed using the operator $\Leftrightarrow$.

A statement of the form $P(x_1, x_2, \ldots, x_n)$ (e.g. $P(x) \Leftrightarrow$ "x is greater than three") is the value of the **propositional function** P at the n-tuple $(x_1, x_2, \ldots, x_n)$, and P is called a **predicate**.

Quantifiers (e.g. universal quantifier $\forall$ and existential quantifier $\exists$) change propositional functions into propositions.

1.3 Sets

A **set** contains a group of similar objects called its **elements** or **members**.

Two sets are equal if and only if they contain the same elements. A set A is said to be a **subset** of another set B if and only if every element in A is also an element of B, denoted by $A \subseteq B$.

If there are exactly n distinct elements in a set S where n is a non-negative number, S is a **finite set**, and n is its **cardinality**. A set is said to be **infinite** if it is not finite.

The **power set** of a set S is the set of all subsets of the set S, denoted by $P(S)$.

An **ordered n-tuple** $(a_1, a_2, \ldots, a_n)$ is the ordered collection that has a_1 as the first element, a_2 as the second element, and so on.

The **Cartesian product** of the sets $A_1, A_2, \ldots, A_n$, denoted by $A_1 \times A_2 \times \ldots \times A_n$, is the set of ordered tuples $(a_1, a_2, \ldots, a_n)$ where $a_i \in A_i$ for $i = 1 \ldots n$.

Most common operations on sets are binary operators **union**, **intersection**, and **difference** denoted by $\cup$, $\cap$, and $\setminus$. The difference of two sets A and B is also called the **complement** of B with respect to A. The complement of a set A, denoted by $\overline{A}$, is the complement of A with respect to the universal set U.

1.4 Integers

If a and b are integers with $a \neq 0$, we say that a **divides** b, denoted by $a|b$, if there is an integer c such that $b = a\dot{c}$. In this case, a is a **factor** of b and that b is a **multiple** of a.

A positive integer p greater than 1 is called **prime** if the only positive factors of p are 1 and p.

For two integers a and b, not both zero, the largest integer d such that $d|a$ and $d|b$ is called the **greatest common divisor** of a and b denoted by $\gcd(a, b)$.

The integers $a_1, a_2, \ldots, a_n$ are **pairwise relatively prime** if $\gcd(a_i, a_j) = 1$ whenever $1 \leq i < j \leq n$.

The **least common multiple** of the positive integers a and b, denoted by $lcm(a, b)$, is the smallest positive integer that is divisible by both a and b.

Let a be an integer and m be a positive integer. We denote by a **mod** m the remainder when a is divided by m.

1.4.1 The Well Ordering Principle and the Mathematical Induction

Every nonempty subset of the set of positive integers contains a smallest element. This is often expressed by saying the set of positive integers is *well ordered*.

Mathematical Induction Principle: Let $S(n)$ denote an open mathematical statement (or set of such open statements) that involves one or more occurences of the variable n, which represents a positive integer.

(a) If $S(1)$ is true; and

(b) If whenever $S(k)$ is true (for some particular, but arbitrarily chosen positive integer k), then $S(k+1)$ is true;

then $S(n)$ is true for all positive integers n.

1.5 Functions

A **function** f from a set A to a set B is an assignment of a unique element of B to each element of A, denoted by $f : A \to B$. Sets A and B are called the **domain** and **codomain** of f, respectively. If $f(a) = b$, we say that b is the **image** of a and a is the **pre-image** of b. The **range** of f is the set of all images of elements of A.

A function f is said to be **one-to-one**, or 1-1, or injective, if and only if $f(x) = f(y)$ implies that $x = y$ for all x and y in the domain of f.

A function f from A to B is called **onto**, or surjective, if and only if for every element $b \in B$ there is an element $a \in A$ with $f(a) = b$.

Most popular operations on functions are **inverse** and **composition**.

A **sequence** is a function from a subset of the set of integers to a set S. We use the notation a_n to denote the image of the integer n. We call a_n a **term** of the sequence.

The symbol $\sum$ is used to denote the **summation** of terms in a sequence. For instance,

$$\sum_{j=m}^{n} a_j$$

represents

$$a_m + a_{m+1} + \ldots a_n.$$

Let f and g be functions from the set of integers or the set of real numbers to the set of real numbers. We say that $f(x) = O(g(x))$ if there are constants C and k such that

$$|f(x)| \leq C|g(x)|$$

whenever $x > k$. We also say that $f(x)$ is $O(g(x))$.

2 The Foundations: Algorithms and Matrices

2.1 Algorithms

An **algorithm** is a definite procedure for solving a problem using a finite number of steps.

Language independent formalization of an algorithm is usually done with the help of an intermediate language between English and a programming language called **psuedocode**.

Algorithms generally share the properties of input, output, definiteness, finiteness, effectiveness, and generality.

The level of efficiency and effectiveness of an algorithm can be analyzed and expressed with the help of its **computational complexity**. An analysis of the time required to solve a problem of a particular size involves **time complexity** of the algorithm. An analysis of the computer memory required involves the **space complexity** of the algorithm.

Complexity analysis of an algorithm is usually done with respect to the **worst case**. Another important type is the **average-case** analysis.

For an algorithm for a problem of input size of n, $O(n)$, $O(\log n)$, and $O(b^n)$ where $b > 1$ time complexities are referred to as **linear**, **logarithmic**, and **exponential**.

2.2 Matrices

A **matrix** is a rectangular array of numbers. A matrix with m rows and n columns is called an $m \times n$ matrix.

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix}$$

Element a_{ij} is the number in ith row and jth column of a matrix A.

Let $A = [a_{ij}]$ and $B = [b_{ij}]$ be $m \times n$ matrices. The **sum** of A and B denoted by $A + B$, is the $m \times n$ matrix that has $a_{ij} + b_{ij}$ as its (i, j)th element. That is, $A + B = [a_{ij} + b_{ij}]$.

Let A be an $m \times k$ matrix and B be a $k \times n$ matrix. The **product** of A and B, denoted by AB, is the $m \times n$ matrix with (i, j)th entry equal to the sum of the products of the corresponding elements from the ith row of A and jth column of B. In other words, if $AB = [c_{ij}]$, then $c_{ij} = a_{i1}b_{1j} + a_{i2}b_{2j} + \ldots + a_{ik}b_{kj}$

The **transpose** of an $m \times n$ matrix $A = [a_{ij}]$, denoted by A^t, is the $n \times m$ matrix obtained by interchanging the rows and columns of A. If $A^t = [b_{ij}]$, then $b_{ij} = a_{ji}$ for $i = 1, 2, \ldots, n$, and $j = 1, 2, \ldots, m$.