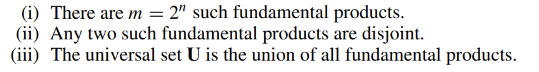
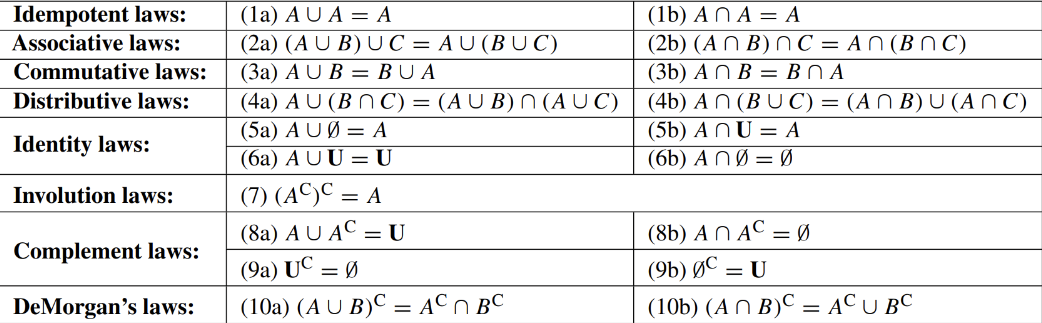
Discrete structures theory base:

* Sets-collection of objects
* Specifying sets can be done by comma or using properties and characteristics of elements.
* Subsets-set where all its elements are contained in another set.
* Every set is its own subset. A is a subset of A.
* N-natural, Z-int, Q-rational, R-real, C-complex
* Universal set contains all objects and sets.
* Empty set has no elements. |Ө|=0, |{Ө}|=1. Empty set is a subset of any set.
* Disjoint sets-have no elements in common.
* Cardinality-| |, lentgh
* Union-
* Intersection - 
* Absolute Complement of a set A – belongs to U, but not A. 
* Relative complement or Difference of a set B with respect to A (MINUS)– belongs to A, but not B. 
* Symmetric difference of A and B(XOR) – belongs to A or B, but not both.



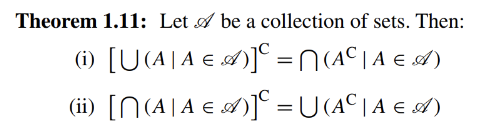
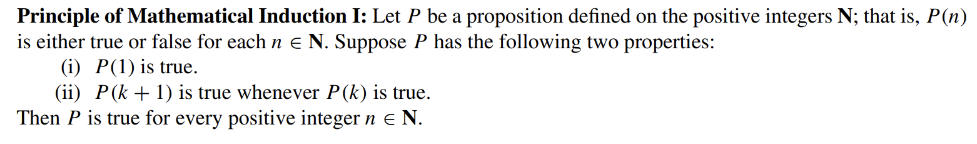
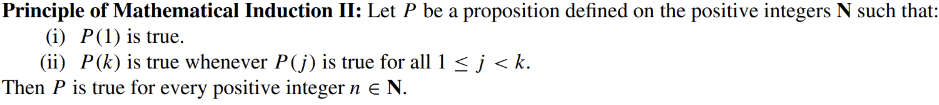
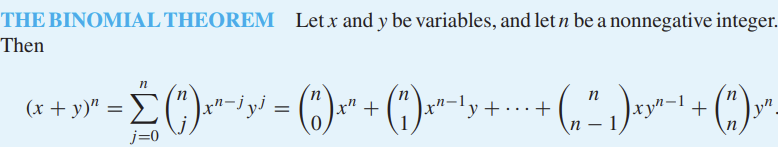
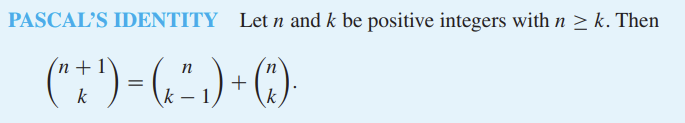
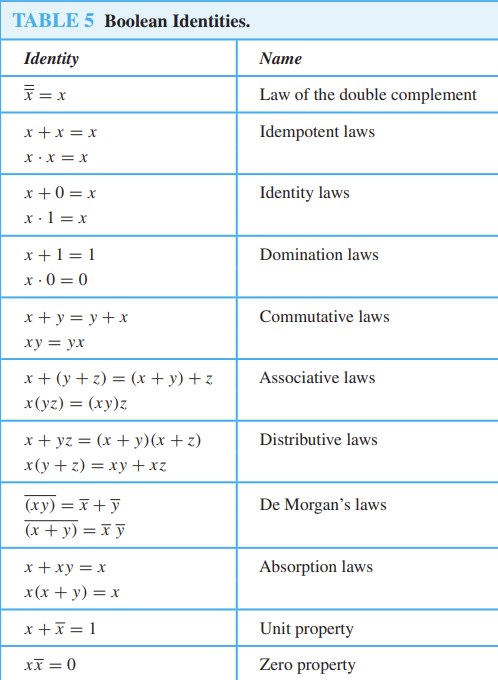
* Power sets(P(s)) – all possible sets of set A. 
* If set has n elements, Power set has 2^n elements
* The ordered n-tuple (a1, a2,...,an) - ordered collection that has a1 as its first element, a2 as its second element,..., and an as its nth element
* Ordered pair – ordered 2-tuple
* The Cartesian product of A and B (A × B) - set of all ordered pairs(order matters) (a, b), where a ∈ A and b ∈ B. So, A × B = {(a, b) | a ∈ A ∧ b ∈ B}.
* Fundamental product - 





* Countable – if set is finite or can be an sequence.
* Inclusion-Exclusion Principle\theorem - A,B are not disjoint
* Partition of S – nonempty subsets of S that don’t overlap



* De Morgan’s law: 
* 
* 
* R is a relation from a set A to a set B and S a relation from B to a set C. The composite of R and S is the relation consisting of ordered pairs (a, c), where a ∈ A, c ∈ C. We denote the composite of R and S by S ◦R
* A relation 𝑅 on 𝑆 is an equivalence relation if 𝑅 is reflexive, symmetric, and transitive.
* A relation 𝑅 on a set 𝑆 is called a partial ordering or a partial order of 𝑆 if 𝑅 is reflexive, antisymmetric, and transitive.
* One-to-one function/injective – if for every x, there is only one y.
* Onto function/subjective– if there exists x for all y.
* One-to-one correspondence/bijective– when its onto and one-to-one
* A function 𝑓: 𝐴 → 𝐵 is invertible if its inverse relation 𝑓 −1 is a function from 𝐵 to 𝐴
* The set of all integers congruent to int a mod m is congruence class of a mod m.
* An integer p greater than 1 is called prime if the only positive factors of p are 1 and p. A positive integer that is greater than 1 and is not prime is called composite
* THE FUNDAMENTAL THEOREM OF ARITHMETIC Every integer greater than 1 can be written uniquely as a prime or as the product of two or more primes where the prime factors are written in order of nondecreasing size
* If n is a composite integer, then n has a prime divisor less than or equal to √n
* THE PRIME NUMBER THEOREM The ratio of the number of primes not exceeding x and x/ ln x approaches 1 as x grows without bound. (Here ln x is the natural logarithm of x.)
* Let a and b be positive integers. Then ab = gcd(a, b) · lcm(a, b)
* Least common multiple: smallest common dividend of a and b
* Linear congruences where ax ≡ b (mod m)
* How many functions are there from a set with 𝑚 elements to a set with 𝑛 elements? 𝑛^m
* How many one-to-one functions are there from a set with 𝑚 elements to one with 𝑛 elements?
  + If m>n, there is no one—to-one functions
  + If m<=n, 𝑛(𝑛 − 1)(𝑛 − 2) … (𝑛 − 𝑚 + 1)
* THE SUBTRACTION RULE. If a task can be done in either 𝑛1 ways or 𝑛2 ways, then the number of ways to do the task is 𝑛1 + 𝑛2 minus the number of ways to do the task that are common to the two different ways
* THE PIGEONHOLE PRINCIPLE If 𝑁 objects are placed into 𝑘 boxes, then there is at least one box containing at least 𝑁/𝑘 objects
* Permutation is an ordered arrangement of objects
* Combination an unordered selection of r elements from the set
* 𝐶(𝑛, 𝑟) is also denoted by  and is called a binomial coefficient.
* Сombinatorial proof is a proof that uses counting arguments to prove that both sides of the identity count the same objects but in different ways or a proof that is based on showing that there is a bijection between the sets of objects counted by the two sides of the identity. These two types of proofs are called double counting proofs and bijective proofs, respectively.
* 
* 
* A proposition is a declarative sentence (that is, a sentence that declares a fact) that is either true or false, but not both
* Propositional variables (or statement variables) variables that represent propositions
* The negation of p means not p
* Conjunction ^ - true when both true
* Disjunction(inclusive or) false when both false
* The conditional statement/implication p → q is the proposition “if p, then q.” The conditional statement p → q is false when p is true and q is false, and true otherwise.
* Exclusive or –
* Converse of  is 
* Contrapositive of  is 
* Inverse of  is 
* Equivalent when compound propositions always have the same value
* Biconditional statement p if and only f q. when tey have the same values
* 𝑝 ↔ 𝑞 is (𝑝 → 𝑞) ∧ (𝑞 → 𝑝)
* The notation 𝑝 ≡ 𝑞 denotes that 𝑝 and 𝑞 are logically equivalent.
* Predicate is expression of one or more variables defined on some domain
* Quantifier is used to quantify the variables of predicates
* Lecture #8 and #9:
* Boolean variable – variables where values can be only 1 or 0
* Boolean function of degree k- function from B^n to B
* Number of Boolean functions of degree n - 
* 
* Lattice – partially ordered set where every pair has a least upper bound and greatest lower bound.
* Literal – Boolean variable or its complement
* Minterm – product of n literals.
* Sum-of-products – sum of minterms, disjunctive normal form
* Maxterm – sum of n literals
* Product-of-sums – product of maxterms, conjuctive normal form
* Functionally complete – {\*, +, -}, {\*, -}, {+,-}, {}, { | }
* Sheffers strelka – “ | “, NAND, 1|1=0
* Pierces arrow -  , NOR, 00=1
* Gates – basic elements of circuits
* AND – округленный кончик, OR – изогнутый и кончик острый, NOT – как треугольник с точкой на носу
* Half adder – adds 2 bits, doesn’t consider previous carry. C=xy, S=x(xor)y
* Full adder- adds bits and carries