CS 131 – Combinatoric Structures
Discussion Module Summary

September 28, 2023

Each discussion worksheet will involve some questions about the lecture material and will be closely related to the problem set. After each session, the solutions to the discussion sheet will be posted on BlackBoard.

Module 1 Summary

1. **Proposition** is a declarative sentence and cannot be interrogative or imperative.
2. **Propositional variables** (or sentential variables) are used to represent propositions in Logic.
   e.g. p = Toronto is the capital of Canada.
3. **Compound propositions** are propositions combined with **logical operators**. Examples of logical operators are:
   binary operators: *conjunction* ($\wedge$), *disjunction* ($\vee$, aka *inclusive or*), *exclusive or* ($\oplus$ - "either - or - but not both")
   unary operators: *negation* ($\neg$).
4. **Truth Tables** are operation tables for logical operators. A truth table shows the truth value of a compound proposition for every possible combination of involved variables.
5. **Implication** ($\rightarrow$): conditional statement - "if ..., then ...", i.e. "if p, then q" is equivalent to $p \rightarrow q$. More statement, please refer to Module 1.8 More Logical Operations, Remarks.
   q if p: $p \rightarrow q$
   q only if p: $q \rightarrow p$
   p if and only if q (*bi-implication, bi-conditional, iff*): $p \leftrightarrow q$
6. In programming (python), we have
   $p \wedge q$: p and q
   $p \vee q$: p or q
   $\neg p$: not p
   $p \oplus q$: $!=$
   $p \rightarrow q$: $\leq$
   $p \leftrightarrow q$: ==
7. **Logic Equivalence** (*equiv* $\equiv$): two propositions have the same truth table. e.g. $p \rightarrow q \equiv \neg p \vee q$
8. **Contraposition**: $p \rightarrow q \equiv (\neg q \rightarrow \neg p)$
9. **Converse**: converse of $p \rightarrow q$ is $q \rightarrow p$
10. **Inverse**: inverse of $p \rightarrow q$ is $\neg p \rightarrow \neg q$
    Note that: the converse and inverse of an implication are equivalent.

11. Proof methods: (1) Truth tables (2) Laws of propositional logic.

12. Conditional identity. $p \to q \equiv \neg p \vee q$

13. **De Morgan Laws**.

$$\neg(p \wedge q) \equiv \neg p \vee \neg q$$

$$\neg(p \vee q) \equiv \neg p \wedge \neg q$$

14. (Read Module **1.11**) **Laws of propositional logic**. In the equivalence table, T denotes the compound proposition that is always true and F denotes the compound proposition that is always false.

Logical Equivalences

Equivalence	Name
$p \wedge T \equiv p$ $p \vee F \equiv p$	Identity Laws
$p \vee T \equiv T$ $p \wedge F \equiv F$	Domination Laws
$p \vee p \equiv p$ $p \wedge p \equiv p$	Idempotent Laws
$\neg\neg p \equiv p$	Double negation Law
$p \wedge q \equiv q \wedge p$ $p \vee q \equiv q \vee p$	Commutative Laws
$p \wedge (q \vee l) \equiv (p \wedge q) \vee (p \wedge l)$ $p \vee (q \wedge l) \equiv (p \vee q) \wedge (p \vee l)$	Distributive Laws
$p \wedge (q \wedge l) \equiv (p \wedge q) \wedge l \equiv p \wedge q \wedge l$ $p \vee (q \vee l) \equiv (p \vee q) \wedge l \equiv p \vee q \ \vee l$	Associative Laws
$\neg(p \wedge q) \equiv \neg p \vee \neg q$ $\neg(p \vee q) \equiv \neg p \wedge \neg q$	De Morgan's Laws
$p \vee (p \wedge q) \equiv p$ $p \wedge (p \vee p) \equiv p$	Absorption Laws
$\neg T \equiv F$ $\neg F \equiv T$	Negation Laws
$p \wedge \neg p \equiv F$ $p \vee \neg p \equiv T$	Complement Laws

15. **Rules of inference**. Please refer to **Module 1.12**.

(Optional Notes)

Definition 1. A *proposition* is a declarative sentence (that is, a sentence that declares a fact) that is either true or false, but not both.

Definition 2. The *truth value* of a proposition is true, denoted by T, if it is a true proposition, and the truth value of a proposition is false, denoted by F, if it is a false proposition.

Definition 3. Let p be a proposition. The *negation* of p, denoted by $\neg p$ (also denoted by $\bar{p}$), is the statement: "It is not the case that p." The truth value of the negation of p, $\neg p$, is the opposite of the truth value of p.

Definition 4. Let p and q be propositions. The *conjunction* of p and q, denoted by $p \wedge q$, is the proposition "p and q." The conjunction $p \wedge q$ is true when both p and q are true and is false otherwise.

Definition 5. Let p and q be propositions. The *disjunction* of p and q, denoted by $p \vee q$, is the proposition "p and q." The conjunction $p \vee q$ is false when both p and q are false and is true otherwise.

Truth Table for Conjunction/Disjunction of two propositions

p	q	$p \wedge q$	$p \vee q$
T	T	T	T
T	F	F	T
F	T	F	T
F	F	F	F

Definition 6. Let p and q be propositions. The *exclusive or* of p and q, denoted by $p \oplus q$, is the proposition that is true when exactly one of p and q is true and is false otherwise.

Truth Table for Exclusive Or of two propositions

p	q	$p \oplus q$
T	T	F
T	F	T
F	T	T
F	F	F

Definition 7. Let p and q be propositions. The *conditional statement* of p and q, denoted by $p \rightarrow q$, is the proposition "if p, then q." The conditional statement $p \rightarrow q$ is false when p is true and q is false, and true otherwise. In the conditional statement $p \rightarrow q$, p is called the *hypothesis* (or *antecedent* or *premise*) and q is called the *conclusion* (or *consequence*).
A conditional statement is also called *implication*.
$p \rightarrow q$ is equivalent to the following statements:
"p is sufficient for q." "q is necessary for p."

Truth Table for Cond. STMT of two propositions

p	q	$p \rightarrow q$
T	T	T
T	F	F
F	T	T
F	F	T

Definition 8. Let p and q be propositions. The *biconditional statement* of p and q, denoted by $p \leftrightarrow q$, is the proposition "p if and only if q." The biconditional statement $p \leftrightarrow q$ is true when p and q have the same truth values, and false otherwise. Biconditional statements are also called *bi-implications*.

Truth Table for biconditional statement of two propositions

p	q	$p \leftrightarrow q$
T	T	T
T	F	F
F	T	F
F	F	T

Definition 9. A compound proposition that is always true, no matter what the true values of the propositional variables that occur in it, is called a *routology*. A compound proposition that is always false is called a *contradiction*. A compound proposition that is neither a tautology nor a contradiction is called a *contingency*.

Examples of Tautology and Contradiction

p	$\neg p$	$p \vee \neg p$	$p \wedge \neg p$
T	F	T	F
F	T	T	F

Definition 11. Compound propositions that have the same truth values in all possible cases are called *logically equivalent*.

The compound propositions p and q are called *logically equivalent* if $p \leftrightarrow q$ is a tautology. The notation $p \equiv q$ that denote p and q are logically equivalent.

Use Truth Tables to prove two statements are logically equivalent.

Module 2 Summary

1. **Boolean algebra** is a way to encode logical arguments into a language that could be manipulated and solved mathematically.

2. Comparing Boolean Algebra and Propositional Logic.

Propositional Logic	False	True	Conjunction $\wedge$	Disjunction $\vee$	Negation $\neg x$
Boolean Algebra	0	1	Boolean Multiplication $\cdot$	Boolean Addition $+$	Boolean $\bar{x}$

Comparison

Law of Propositional Logic	Boolean Algebra	Name
$p \wedge T \equiv p$ $p \vee F \equiv p$	$p \cdot 1 = p$ $p + 0 = p$	Identity Laws
$p \vee T \equiv T$ $p \wedge F \equiv F$	$p + 1 = 1$ $p \cdot 0 = 0$	Domination Laws
$p \vee p \equiv p$ $p \wedge p \equiv p$	$p + p = p$ $p \cdot p = p$	Idempotent Laws
$\neg\neg p \equiv p$	$\bar{\bar{p}} = p$	Double negation Law
$p \wedge q \equiv q \wedge p$ $p \vee q \equiv q \vee p$	$p \cdot q = q \cdot p$ $p + q = q + p$	Commutative Laws
$p \wedge (q \vee l) \equiv (p \wedge q) \vee (p \wedge l)$ $p \vee (q \wedge l) \equiv (p \vee q) \wedge (p \vee l)$	$p \cdot (q + l) = p \cdot q + p \cdot l$ $p + (q \cdot l) = (p + q) \cdot (p + l)$	Distributive Laws
$p \wedge (q \wedge l) \equiv (p \wedge q) \wedge l \equiv p \wedge q \wedge l$ $p \vee (q \vee l) \equiv (p \vee q) \wedge l \equiv p \vee q \ \vee l$	$(p \cdot q) \cdot l = p \cdot (q \cdot l)$ $(p + q) + l = p + (q + l)$	Associative Laws
$\neg(p \wedge q) \equiv \neg p \vee \neg q$ $\neg(p \vee q) \equiv \neg p \wedge \neg q$	$\overline{p \cdot q} = \bar{p} + \bar{q}$ $\overline{p + q} = \bar{p} \cdot \bar{q}$	De Morgan's Laws
$p \vee (p \wedge q) \equiv p$ $p \wedge (p \vee p) \equiv p$	$p + (p \cdot q) = p$ $p \cdot (p + q) = p$	Absorption Laws
$\neg T \equiv F$ $\neg F \equiv T$	$\bar{1} = 0$ $\bar{0} = 1$	Negation Laws
$p \wedge \neg p \equiv F$ $p \vee \neg p \equiv T$	$p \cdot \bar{p} = 0$ $p + \bar{p} = 1$	Complement Laws

3. Two Boolean expressions are equivalent if they have the same truth tables.

4. Binary and denary conversion. **Binary system** is combination of 0s and 1s. **Decimal (denary) system** is combination of 10 unique digits 0 to 9.

   Suppose we have a binary number $a = i_n i_{n-1} \ldots i_1 i_0$ where i_k is 0 or 1, we can convert a as $\sum_{k=0}^{n} i_k \cdot 2^k$ to denary number.

5. **minterm**: a conjunction of all input-bits where some of the input-bits can be negated but the negation should be applied on individual input-bits.

6. **clause**: a disjunction of all or some input-bits where some of the input-bits can be negated but the negation should be applied on individual input-bits.

7. **DNF formula**: (Disjunctive normal form) OR of ANDs.

8. **CNF formula**: (Conjunctive normal form) And of ORs.

9. **Logic circuit**: the algorithm that computer engineers use to make an electronic device.

10. **Logic gates**: (images refer to Module 2.10)

    AND gate: Boolean multiplication,

    OR gate: Boolean addition,

    inverter (NOT gate): Boolean complement.

11. More definitions and vocabulary introduced in Module 2, please refer to **Module 2.12**.

Discussion Problems. See Discussion Sheet on BlackBoard.

Module 3 Summary

1. **Set**: A *set* is an unordered collection of distinct objects forming a group. Objects in a set are called *elements or members of the set*.
2. **Finite sets**: sets that have a finite number of elements. **Infinite sets**: sets that have infinitely many elements. Commonly used infinite sets include:

   $\mathbb{N}$: set of all natural numbers $\{0,1,2...\}$

   $\mathbb{Z}^+$: set of all positive integers $\{1,2,3...\}$

   $\mathbb{Z}$: set of all integers $\{...,-2,-1,0,1,2,...\}$

   $\mathbb{Q}$: set of all rational numbers $\{x = \frac{a}{b} | a,b \in \mathbb{Z}, b \neq 0\}$

   $\mathbb{R}$: set of all real numbers
3. **Cardinality**: the length of a finite set A.
4. **Empty set** (null set): a set with no element, denoted as $\emptyset$. **The cardinality of the empty set $\emptyset$ is zero.**
5. If an element a belongs to a set A, we have $a \in A$; if a is not in A, we have $a \notin A$
6. $\in$ and $\notin$ are Boolean operations - their output is either True or False.
7. $A \cup B$ **(union)** and $A \cap B$ **(intersection)** of two sets A and B will create a new set, defined by:

   $A \cup B = \{x | x \in A \vee x \in B\}$ builds a new set of all elements that are elements of A OR B.

   $A \cap B = \{x | x \in A \wedge x \in B\}$ builds a new set of all elements that are elements of A AND B.

   *For Venn diagram for union and intersection, please refer to Module 3.5.
8. More set operations:

   $\bar{A} = A^c = \{x | x \notin A\}$: complement set

   $A - B = A \setminus B = \{x | x \in A \wedge x \notin B\}$: set difference

   $A \oplus B = \{x | x \in A \oplus x \in B\}$
9. $A \subseteq B$: A is a subset of B. $\forall x \in A, x \in B$
10. $A \nsubseteq B$: A is not a subset of B. $\exists x \in A, x \notin B$
11. **Predicate** (Propositional function) $P(x)$: the value of a proposition p depends on the value of x.
12. Domain is called the domain of discourse or the universe of discourse.
13. A predicate which is bound by a quantifier is called a quantified statement.
14. **Universal quantifier** ($\forall$): for all. $\forall x \in U(P(x))$: for all values of x in the domain, $P(x)$ is true.
15. **Existential quantifier** ($\exists$): there exists. $\exists x \in U(P(x))$: there is at least one value of x in the domain for which $P(x)$ is true.
16. De Morgan's laws for quantifiers:

    $\neg \forall x \in U(P(x)) \equiv \exists x \in U(\neg P(x))$. There is an x for which $P(x)$ is false.

    $\neg \exists x \in U(P(x)) \equiv \forall x \in U(\neg P(x))$. For all x, $P(x)$ is false.
17. Empty set is a subset of all sets.

18. Proof methods in sets:

    Set builder notation: To prove that two sets are equal each other (e.g.$A = B$), we must start with one of the sets and transform it into the other set using a sequence of steps by applying set builder notation.

    Set identities: The same as the laws of propositional logic, there are some set identities (laws for sets) which are proved to be true. To prove that two sets are equal each other (e.g. $A = B$), we can use these basic set identities.

    Two-column proof format: Proving statements using a sequence of steps by applying the laws of logic, rules of inference and set definitions in a two-column proof format.

    Paragraph proof format: (using definitions) Proving statements using a sequence of steps by applying the laws of logic, rules of inference and set definitions in a paragraph proof format.

19. Set Identity table.

Set Identity	Name
$A \cap U = A$ $A \cup \emptyset = A$	Identity Laws
$A \cup U = U$ $A \cap \emptyset = \emptyset$	Domination Laws
$A \cup A = A$ $A \cap A = A$	Idempotent Laws
$\bar{\bar{A}} = A$	Double negation Law
$A \cup B = B \cup A$ $A \cap B = B \cap A$	Commutative Laws
$A - B = A \cap \bar{B}$	Subtraction Laws
$A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$ $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$	Distributive Laws
$A \cup (B \cup C) = (A \cup B) \cup C$ $A \cap (B \cap C) = (A \cap B) \cap C$	Associative Laws
$\overline{A \cap B} = \bar{A} \cup \bar{B}$ $\overline{A \cup B} = \bar{A} \cap \bar{B}$	De Morgan's Laws
$A \cup (A \cap B) = A$ $A \cap (A \cup B) = A$	Absorption Laws
$A \cup \bar{A} = U$ $A \cap \bar{A} = \emptyset$	Complementation Laws

20. A quantified statement with multiple quantifier is said to have **nested quantifiers**. $\forall x$ and $\exists y$ mean for all x in the domain of discourse and there exist y in the domain of discourse respectively.

Discussion Problems. See Discussion Sheet on BlackBoard.

Module 4 Summary

1. The set product (Cartesian product) of sets A and B, denoted by $A \times B$, is the set of all ordered pairs (a,b), where $a \in A$ and $b \in B$.
$$A \times B = \{(a,b) | a \in A \land b \in B\}$$
e.g. $\mathbb{Z} \times \mathbb{Z}$: two dimensional Cartesian coordinates, includes all 2D points (x,y) where x,y are integers. Also denoted by $\mathbb{Z}^2$.

2. Cartisian products $A \times B$ and $B \times A$ are not equal unless $A = \emptyset$ or $B = \emptyset$ or $A = B$.

3. $A^n = A_1 \times A_2 \times ... \times A_n = \{(a_1, a_2, ..., a_n) | a_i \in A_i \text{ for } i = 1, 2, ..., n\}$

4. $\mathbb{Z}^3 = \mathbb{Z} \times \mathbb{Z} \times \mathbb{Z} = \{(x,y,z) | x,y,z \in \mathbb{Z}\}$ includes all 3D points. For figure please refer to Module 4.4.

5. **Relation**. Any subset of the set product $A \times B$ is called a **relation** from the set A to the set B. The elements of relations are ordered pairs, where the first element belongs to A and the second to B (Relations are sets of ordered pairs.)

   e.g. functions, injection, surjection, bijection

6. Set R is a (binary) relation between the sets X and Y if $R \subseteq X \times Y$

7. A relation from a set A to itself is called a relation on set A.

   e.g. reflexive, symmetric relation, transitive relation, equivalence

8. Set R is a (binary) relation on a set X if $R \subseteq X \times X$.

9. **Function**. Function R is a relation between sets X and Y if $\forall x \in X, \exists! y \in Y, (x,y) \in R$, denoted by $R : X \to Y$.

   Domain: set X is the domain of R.

   Co-domain: set Y the target (co-domain) of R.

   Range/Image: an element $y \in Y$ is in the range of R when there is $x \in X$ s.t. $(x,y) \in R$. Set builder notation for range of R is: $\{y \in Y | (x,y) \in R \text{ for some } x \in X\}$.

   Two functions are **equal** if their domain and target are equal, and the image of both functions is the same for each element in the domain.

10. **Injection** (one-to-one). Relation R is an injection iff:

    1) R is a function from X to Y i.e. $\forall x \in X, \exists! y \in Y, (x,y) \in R$.

    2) $\forall y \in Y$, there is at most one $x \in X$ s.t. $(x,y) \in R$

11. **Surjection** (onto). Relation R is a surjection iff:

    1) R is a function from X to Y i.e. $\forall x \in X, \exists! y \in Y, (x,y) \in R$.

    2) $\forall y \in Y$, there is at least one $x \in X$ s.t. $(x,y) \in R$.

12. **Bijection**. Relation R is a bijection if it is both injection and surjection.

13. Suppose R is a relation defined on A. Such relation is a subset of $A \times A$.

    Reflexive. R is reflexive if $\forall x \in A, (x,x) \in R$

    Anti-reflexive. R is anti-reflexive if $\forall x \in A, (x,x) \notin R$

14. An element x is related to element y (i.e. $(x,y) \in R$), we have

    prefix notation: $R(x,y)$

    infix notation: xRy.

15. Suppose that relation R is defined on set A.

- R is reflexive if $\forall x \in A, xRx$. (repeat from previous def and change notation)
- R is **symmetric** if $\forall x, y \in A, xRy \rightarrow yRx$
- R is **anti-symmetric** if $\forall x, y \in A, xRy \rightarrow$ not yRx where $x \neq y$.
  That is, for all $x, y \in A$, if xRy with $x \neq y$, then yRx must not hold. Equivalently, if xRy and yRx, then $x = y$.
- R is **transitive** if $\forall x, y, z \in A, (xRy \text{ and } yRz) \rightarrow xRz$
- R is **equivalence** if it is reflexive, symmetric and transitive.
- R is **permutation** if it is a bijection between A and A.

16. When x divides y (or y is divisible by x, denoted by $x|y$), then there exists a number q s.t. $x \cdot q = y$, i.e. $x|y \equiv \exists q \in \mathbb{Z}(y = x \cdot q)$
17. For formal proof and informal proof (paragraph form), please see examples in Module 4.7.
18. A **partition** of a nonempty set A is a set of nonempty subsets of A such that the subsets are pairwise disjoint and union of the subsets makes A. $P = \{A_1, A_2, ..., A_n\}$ is a partition for A if

    (1) $\forall i, A_i \subseteq A$.

    (2) $\forall i, A_i \neq \emptyset$.

    (3) $\{A_1, A_2, ..., A_n\}$ are pairwise disjoint, i.e, $\forall i, j \in S, \{1\} A_i \cap A_j = \emptyset$ where $S = \{i, j | 1 \leq i, j \leq n \wedge i \neq j\}$.

    (4) $A = A_1 \cup A_2 \cup ... \cup A_n$
19. If x and y are in the same part of the a partition then they are related, i.e.xRy. Otherwise, there shuold be no relation between x and y, i.e. not xRy
20. Theorem: R on A is equivalence $\rightarrow R$ partitions A.

Discussion Problems. See Discussion Sheet on BlackBoard.

Module 5 Summary

1. Recommended reading: Module 5.4 **Guide to Proof-Writing in a Paragraph Form**.
2. Proof methods: **direct proof, examples and counterexamples, contrapositive proof, proof by contradiction, proof by cases and exhaustive proof.**
3. **Fermat's Last Theorem** (Fermat's Conjecture):

$$\forall a > 0 \ \forall b > 0 \ \forall c > 0 \ \forall n > 2 \ (a^n + b^n \neq c^n)$$

4. **Contrapostive Proof**. Logical equivalence gives $p \rightarrow q \equiv \neg q \rightarrow \neg p$. Therefore, to prove "if p then q", we can alternatively prove "if $\neg q$, then $\neg p$".
5. **Proof by Contradiction**. Logical equivalence gives $p \rightarrow q \equiv (p \wedge \neg q \rightarrow \text{False})$. Moreover, we can choose to prove $p \wedge \neg q \rightarrow r \wedge \neg r$. In general, if we want to prove $p \rightarrow q$, we can do: For the sake of contradiction, we assume that $\neg q$ is true, and prove that $\neg q$ implies some situations that violate a true fact or theorem. Therefore, by contradiction, we know that $p \rightarrow q$ must be true.
6. **Exhaustive proof**: test all elements in a domain. **Proof by cases**: divide the domain of a problem into some sub domains and prove the problem in each sub domain.
7. Q: how to decide which proof method to use? First try if direct proof is straight forward; if not, try alternative strategies, such as contoposition, contradiction, etc.
8. Theorem: Suppose x, y, z are integers. If $x|y$ and $x|z$, then $x|(sy + tz)$ for any integers s, t.
9. **Division Theorem** (Division Algorithm): Suppose n is an integer and d is a postive integer. Then, there exist unique integer values q and r such that $n = d \cdot q + r$, with $0 \leq r \leq d - 1$
10. A **prime numbe**r is a number which has exactly two divisors i.e., 1 and the number itself. Examples are 2, 3, 5, 7 and 11.
11. A **composite number** has more than two divisors, which means apart from getting divided by 1 and the number itself, it can also be divided by at least one positive integer. One example is 6 because divisors(6) = $\{1,2,3,6\}$. A composite number n can be written as product of two integers $n = a \cdot b$ such that $1 < a < n$ and $1 < b < n$.
12. 1 is neither a prime nor a composite number.
13. Every natural number (except 1) has at least 2 divisors.
14. Every natural number except 1 is divisible by some prime number.
15. **Well-ordering principle**: any subset of natural numbers contains a smallest element.
16. A number is **irrational** if it cannot be written as a fraction.
17. $\sqrt{2}$ is irrational.

Discussion Problems. See Discussion Sheet on BlackBoard.

Module 6 Summary

1. A sequence is a special type of function with a domain of $0,1,2,\ldots$ or $1,2,3,\ldots$. We use the notation a_n to denote the image of the index n that represents the term of a sequence. E.g. $a_3 = 0.35$ means that index 3 of the sequence is 0.35.

2. Some (but not all) sequences can be written in a closed form formula (or a function format).

3. $\sum_{i=1}^{n} i = 1+2+...+n = \frac{n(n+1)}{2}$

4. **Arithmetic progression** (AP) or arithmetic sequence starts with an initial term and after the initial term, each term equals the previous term added by a fixed number called the common difference. In general, arithmetic sequence is written as function $z_n = a + nd$ where $n \geq 0$, a is the initial term and d is the common difference.

5. **Geometric progression** (GP) or geometric sequence starts with an initial term and after the initial term, each term equals the previous term multiplied by a fixed number called the common ratio. In general, geometric sequence is written as function $a_n = ar^n$, where $n \geq 0$, a is the initial term and r is the common ratio.

6. Every sequence must be expressed via a recurrence relation which means expressing term an as a function of one or more previous terms of the sequence.

7. **Fibonacci sequence** is defined by the initial terms $f_0 = 0, f_1 = 1$ and the recurrence relation $f_n = f_{n-1} + f_{n-2}$, $n \geq 2$.

8. To express the summation of $a_m, a_{m+1}, ..., a_n$, we write $\sum_{k=m}^{n} a_k$. Equality can be found: $\sum_{k=m}^{n} a_k = \sum_{k=0}^{n-m} a_{m+k}$ when manipulating the upper and lower bounds of the index.

9. Chaning the name of the index does not impact the result of a sigma, that is $\sum_{k=m}^{n} a_k = \sum_{i=m}^{n} a_i$

10. To express the product of terms of a sequence $(a_m, a_{m+1}, ..., a_n)$, we write $\prod_{k=m}^{n} a_k = \prod_{k=0}^{n-m} a_{m+k} = a_m \cdot a_{m+1} \cdot \ldots \cdot a_n$.

11. **Mathematical Induction** can be used to prove a statement $P(n)$ holds for every natural number $n = 0,1,2,....$ In predicate logic, we have $(P(1) \wedge \forall k \geq 1(P(k) \rightarrow P(k+1))) \rightarrow \forall n P(n)$.

12. (weak) Mathematical induction proves a predicate $P(n)$ holds for every number n using two steps:
    - Basis step (base case): $P(1)$ is true
    - Inductive step: $P(k) \rightarrow P(k+1)$ is true. That is: Suppose $P(k)$ is true for $k \geq 1$, and prove $P(k+1)$ is true.

13. **Strong Induction**. In strong induction, we assume that $P(n)$ holds for all $n \leq k$, and prove that $P(n)$ holds for $n = k+1$. In predicate logic, we have $\forall k((P(1) \wedge P(2) \wedge ... \wedge P(k)) \rightarrow P(k+1)) \rightarrow \forall n P(n)$

Discussion Problems. See Discussion Sheet on BlackBoard.

Module 7 Summary

1. **The Product Rule**: Let $A_1, A_2, ..., A_n$ be finite sets. Then the cardinality of the product of these finite sets equals the product of the cardinality of the sets, i.e. $|A_1 \times A_2 \times ... \times A_n = |A_1| \cdot ... \cdot |A_n|$

2. **The Sum Rule**: If n sets $A_1, ..., A_n$ are *mutually disjoint*, then the cardinality of the union of the sets equals the summation of the cardinality of the sets, i.e., $|A_1 \cup ... \cup A_n| = |A_1| + ... + |A_n|$

3. **Permutation**: ordered selection of a set of objects

4. **Combination**: unordered selection of a set of objects

5. **Permutation with repetition**: ordered selection of a set of objects that may be identical

6. **Combination with repetition**: unordered selection of a set of objects that may be identical

7. **r-permutation from n objects**: denoted as $P(n,r)$, suppose we have a set of n distinct objects and we want to make a sequence of r objects all taken from the set $r \leq n$. $P(n,r) = \frac{n!}{(n-r)!} = n(n-1)...(n-r+1)$

8. If $n = r$, then $P(n,n) = \frac{n!}{0!} = n!$

9. **r-combination (r-subset) from n objects**: denoted as $C(n,r)$, suppose we have a set of n distinct objects and we want to select r objects all taken from the set $r \leq n$. (when order of selection does not matter) $C(n,r) = \binom{n}{r} = \frac{P(n,r)}{P(r,r)} = \frac{P(n,r)}{r!} = \frac{n!}{r!(n-r)!}$.

10. $\binom{n}{r} = \binom{n}{n-r}$.

    Proof. $\binom{n}{n-r} = \frac{P(n,n-r)}{P(n-r,n-r)} = \frac{P(n,n-r)}{(n-r)!} = \frac{n!}{(n-r)!(n-n+r)!} = \frac{n!}{(n-r)!r!} = \binom{n}{r}$ □

11. Permutation with repetition. Suppose we have n objects (some of them might be identical, say n_1 of object 1, n_2 of object 2, ... , n_k of object k) and we want to make a sequence out of those objects, then the total number of possible arrangements (counting) is $\frac{n!}{n_1!n_2!...n_k!} = \binom{n}{n_1}\binom{n-n_1}{n_2}...\binom{n-n_1-n_2...-n_{k-1}}{n_k}$

12. Combination with repetition. Suppose we have n *identical* balls and m numbered bins. We want to put the balls into the bins where there is no restriction meaning that all balls may go to one bin or they can be evenly distributed between the m bins or any other assignments. Then the number of possible arrangements is $\binom{n+m-1}{m-1}$

13. **Complementary Counting**. Let A be a set and has a complement denoted by $\bar{A}$. Then $|A| = |U| - |\bar{A}|$. (e.g. counting for arrangement containing *at least one* = counting for all arrangement - counting for arrangement that contains zero)

14. **Pigeonhole Principle**. If $n+1$ pigeons are placed in n boxes, then there must be at least one box with more than one pigeon.

15. The **inclusion-exclusion principle** is a counting technique which generalizes the sum rule where the sets are not mutually disjoint. For two sets, the inclusion-exclusion principle is $|A_1 \cup A_2| = |A_1| + |A_2| - |A_1 \cap A_2|$

Discussion Problems. See Discussion Sheet on BlackBoard.
End of Semester!