# Boolean Logic and Sets - Detailed Notes

**Author: Shekh Mostofa Abedin**

## 1. Boolean Logic

### Basic Operations

- **AND ($\wedge$)**:
  - True if both operands are true.
  - Example: $1 \wedge 1 = 1$, $1 \wedge 0 = 0$
- **OR ($\vee$)**:
  - True if at least one operand is true.
  - Example: $1 \vee 0 = 1$, $0 \vee 0 = 0$
- **NOT ($\neg$)**:
  - Inverts the value.
  - Example: $\neg 1 = 0$, $\neg 0 = 1$
- **XOR ($\oplus$)**:
  - True if the operands are different.
  - Example: $1 \oplus 0 = 1$, $1 \oplus 1 = 0$
- **IF-THEN ($\Rightarrow$)**:
  - Also known as implication. $p \Rightarrow q$ is false only when $p$ is true and $q$ is false. In all other cases, it is true.
  - Example: If $p = 1$ (True) and $q = 0$ (False), then $p \Rightarrow q = 0$ (False).
  - If $p = 0$, the result is always True, regardless of $q$.
- **IF AND ONLY IF ($\Leftrightarrow$)**:
  - Also known as biconditional. $p \Leftrightarrow q$ is true if both $p$ and $q$ are either true or false. It is false if one is true and the other is false.
  - Example: If $p = 1$ and $q = 1$, then $p \Leftrightarrow q = 1$. If $p = 1$ and $q = 0$, then $p \Leftrightarrow q = 0$.

### Truth Tables

- A truth table lists all possible values of a Boolean expression.

### Truth Tables for Basic Logical Operators

**1. AND ($\wedge$)**

| p | q | p ∧ q |
|---|---|---|
| T | T | T |
| T | F | F |
| F | T | F |
| F | F | F |

### 2. OR ($\vee$)

| p | q | p $\vee$ q |
|---|---|---|
| T | T | T |
| T | F | T |
| F | T | T |
| F | F | F |

### 3. XOR ($\oplus$)

| p | q | p $\oplus$ q |
|---|---|---|
| T | T | F |
| T | F | T |
| F | T | T |
| F | F | F |

### 4. NOT ($\neg$)

| p | $\neg$p |
|---|---|
| T | F |
| F | T |

### 5. IF-THEN ($\Rightarrow$)

| p | q | p $\Rightarrow$ q |
|---|---|---|
| T | T | T |
| T | F | F |
| F | T | T |
| F | F | T |

### 6. IF AND ONLY IF ($\Leftrightarrow$)

| p | q | p $\Leftrightarrow$ q |
|---|---|---|
| T | T | T |
| T | F | F |
| F | T | F |

| p | q | p ⇔ q |
|---|---|---|
| F | F | T |

**Example**:

The expression we are evaluating is a combination of all the basic operators: `(p ∧ q) ∨ (¬p ⊕ r) ⇒ (p ⇔ q)`

| p | q | r | ¬p | p ∧ q | ¬p ⊕ r | (p ∧ q) ∨ (¬p ⊕ r) | p ⇔ q | (p ∧ q) ∨ (¬p ⊕ r) ⇒ (p ⇔ q) |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 |
| 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 |
| 1 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 |

### Bitwise Operations

- **AND, OR, XOR, NOT** can also be applied to binary numbers bit by bit.
  - **AND Example**: `1101 ∧ 1011 = 1001`
  - **OR Example**: `1101 ∨ 1011 = 1111`
  - **XOR Example**: `1101 ⊕ 1011 = 0110`
  - **NOT Example**: `¬1101 = 0010`
- **Shifts**:
  - **Left Shift (`<<`)**: Shifts bits to the left, filling with zeros on the right.
    - Example: `1011 << 2 = 101100`
  - **Right Shift (`>>`)**: Shifts bits to the right, discarding bits on the right.
    - Example: `1101 >> 2 = 0011`

### Logical Implications

- **Implication (`⇒`)**:
  - `p ⇒ q` is false only when `p` is true and `q` is false.
  - Example: If `p = 1` and `q = 0`, then `p ⇒ q` is false.

---

## 2. Sets

### Overview

## Set Operations

- **Union (∪)**: The set containing all elements from both sets.
  - Example: `A ∪ B` where `A = {1, 2}` and `B = {2, 3}` results in `{1, 2, 3}`.
- **Intersection (∩)**: The set containing only elements that are in both sets.
  - Example: `A ∩ B` where `A = {1, 2}` and `B = {2, 3}` results in `{2}`.
- **Relative Complement (`A - B`)**: The set of elements in `A` that are not in `B`.
  - Example: `A - B` where `A = {1, 2, 3}` and `B = {2, 4}` results in `{1, 3}`.
- **Complement (`A'`)**: The set of elements not in `A` relative to the universal set `U`.
  - Example: If `U = {1, 2, 3, 4, 5}` and `A = {1, 2}`, then `A' = {3, 4, 5}`.

## Set Builder Notation

## Set Builder Notation

- Describes a set by stating the properties that its members must satisfy.
  - **Example**: `A = { x | x ∈ N, x < 5 }` defines the set of natural numbers less than 5, i.e., `{1, 2, 3, 4}`.

**Different Possible Conditions in Set Builder Notation:**

- **Condition on Membership in a Set**:
  - `x ∈ N`: ( x ) is a natural number.
  - `x ∈ Z`: ( x ) is an integer.
  - `x ∈ Q`: ( x ) is a rational number.
- **Conditions Based on Inequalities**:
  - `x < 10`: ( x ) is less than 10.
  - `x ≥ 0`: ( x ) is greater than or equal to 0.
  - `3 ≤ x < 15`: ( x ) is between 3 and 15 (inclusive of 3, but less than 15).
- **Conditions Based on Modular Arithmetic**:
  - `x mod 2 = 0`: ( x ) is even.
  - `x mod 3 = 1`: When divided by 3, ( x ) leaves a remainder of 1.
- **Conditions Based on Divisibility**:
  - `x \ mod \ 5 = 0`: ( x ) is divisible by 5.
  - `x \ mod \ 4 ≠ 0`: ( x ) is not divisible by 4.

**Example Combining Multiple Conditions:**

- **Example**: `A = { x | x ∈ N, x < 10, x mod 2 == 0 }`
  - **Explanation**:
    - ( x ) is a natural number.
    - ( x ) is less than 10.
    - ( x ) is even (i.e., ( x ) is divisible by 2).

    - **Result**: ( A = {2, 4, 6, 8} ).

- **Example**: `B = { y | y ∈ Z, y > -5, y mod 3 = 1 }`
    - **Explanation**:
        - ( y ) is an integer.
        - ( y ) is greater than -5.
        - ( y ) leaves a remainder of 1 when divided by 3.
    - **Result**: ( B = {-4, -1, 2, 5, 8, \dots} ) (continuing with values greater than -5 that satisfy the conditions).

## Venn Diagrams

- **Visual Representation**: Venn diagrams are often used to represent sets and their relationships.
    - **Union**: Area covered by both circles.
    - **Intersection**: Overlapping area of circles.
    - **Complement**: Area outside the circle of the set being considered.

## Key Points to Remember

- **No duplicates in a set**.
- **Order of elements in a set does not matter**.
- **Set operations are foundational in mathematics and computer science**.