Lecture 2

- Logic
  - Compound Statements
  - Conditional Statements
  - Valid & Invalid Arguments
  - Digital Logic Circuits

Reading (Epp's textbook)
2.1 - 2.4

Logic

- Logic is a system based on **statements**.
- A **statement** (or **proposition**) is a sentence that is true or false but not both.
- We say that the truth value of a statement is either true (T) or false (F).
- Corresponds to 1 and 0 in digital circuits
- Examples:
  - "520 < 499"

    Is this a statement? Yes What is the truth value? F
  - "x+ y = 5"

    Is this a statement? No What is the truth value? ?
  - "x < y if and only if y > x"

    Is this a statement? Yes What is the truth value? T

Compound Statements

- One or more statements can be combined to form a single compound statement.
- We formalize this by denoting statements with letters such as p, q, r, s, and introducing several logical operators.
- We will examine the following logical operators:

Logical Operator	Denote	Symbol
Negation	NOT	~ or $\neg$
Conjunction	AND	$\wedge$
Disjunction	OR	$\vee$
Exclusive or	XOR	$\oplus$

The order of operations can be overridden through the use of parentheses.

- These logical operators are used to build more complicated logical expressions out of simpler ones.

Truth Table

- A **statement form** is an expression made up of statement variables (such as p, q, and r) and logical connectives (such as ~, $\wedge$, and $\vee$).
- A **Truth table** for a given statement form displays the truth values that correspond to all possible combinations of truth values for its component statement variables.

Truth table for NOT

p	**~p**
T	F
F	T

Truth table for AND

p	**q**	**p$\wedge$q**
T	T	T
T	F	F
F	T	F
F	F	F

Truth table for OR

p	**q**	**p$\vee$q**
T	T	T
T	F	T
F	T	T
F	F	F

Exclusive or (XOR)

- When or is used in its exclusive sense:

P $\oplus$ q means "p **or** q **but** not both"

$$p \oplus q = (p \vee q) \wedge \sim(p \wedge q).$$

Truth table for XOR

p	**q**	**p∨q**	**p∧q**	**~(p∧q)**	**p ⊕ q**
T	T	T	T	F	F
T	F	T	F	T	T
F	T	T	F	T	T
F	F	F	F	T	F

Truth table for (r ∨ (~(p∧q))?

Logical Equivalence

- Two *statement forms* are called **logically equivalent** if, and only if, they have identical truth values for each possible substitution of statements for their statement variables. The logical equivalence of statement forms *P* and *Q* is denoted by writing $P \equiv Q$.
- **(Showing equivalence)** Are the statements of the forms $\sim(p \wedge q)$ and $\sim p \vee \sim q$ logically equivalent?

p	**q**	**$\sim p$**	**$\sim q$**	**$p \wedge q$**	**$\sim(p \wedge q)$**	**$\sim p \vee \sim q$**
T	T	F	F	T	**F**	**F**
T	F	F	T	F	**T**	**T**
F	T	T	F	F	**T**	**T**
F	F	T	T	F	**T**	**T**

De Morgan's Laws:

$\boldsymbol{\sim(p \wedge q) \equiv \sim p \vee \sim q}$ and $\boldsymbol{\sim(p \vee q) \equiv \sim p \wedge \sim q}$

Tautologies and Contradictions

- A **tautology** is a statement form that is always true. A statement whose form is a tautology is a **tautological statement (denoted by t)**.
- A **contradiction** is a statement form that is always false. A statement whose form is a contradiction is a **contradictory statement (denoted by c)**.

p	**~p**	**p $\vee$ ~p**	**p $\wedge$ ~p**
T	F	T	F
F	T	T	F

Simplifying Statement Forms

$\sim(q \wedge t) \wedge \underline{\sim(\sim q \wedge p)} \wedge (p \vee q)$

- By De Morgan's Law $\equiv \sim(q \wedge t) \wedge (\underline{\sim(\sim q)} \vee \sim p) \wedge (p \vee q)$
- By the Double negative law $\equiv \underline{\sim(q \wedge t)} \wedge (q \vee \sim p) \wedge (p \vee q)$
- By the Identity Law $\equiv \sim q \wedge \underline{(q \vee \sim p) \wedge (p \vee q)}$
- By the Distributive Law $\equiv \sim q \wedge (q \vee \underline{(\sim p \wedge p)})$
- By the Negation Law $\equiv \sim q \wedge \underline{(q \vee c)}$
- By the Identity Law $\equiv \underline{\sim q \wedge q}$
- By the Negation Law $\equiv c$

Conditional statements

- **Implication (if - then):** "If p then q" or "p implies q" and is denoted $p \rightarrow q$. It is false when p is true and q is false; otherwise it is true.
- We call p the hypothesis of the conditional and q the conclusion.

Truth table for p→q

p	q	p→q
T	T	T
T	F	F
F	T	T
F	F	T

Construct a truth table for the statement form $\boldsymbol{p \rightarrow q \equiv \sim p \vee q}$.

Conditional statements

Suppose a conditional statement of the form "If p then q" is given.

- **The Negation** of the conditional statement is logically equivalent to "p and not q."

  Symbolically: $\sim(p \rightarrow q) \equiv p \wedge \sim q.$

- **The Contrapositive** of the conditional statement is "If $\sim q$ then $\sim p$."

  Symbolically: $p \rightarrow q$ is $\sim q \rightarrow \sim p$.

- **The Converse** of the conditional statement is "If q then p."

  Symbolically: $p \rightarrow q$ is $q \rightarrow p$.

- **The Inverse** of the conditional statement is "If $\sim p$ then $\sim q$."

  Symbolically: $p \rightarrow q$ is $\sim p \rightarrow \sim q$.

Conditional statements

- **Biconditional (if and only if):** "p if, and only if, q" and is denoted $p \leftrightarrow q$. It is true if both *p* and *q* have the same truth values and is false if *p* and *q* have opposite values.

Truth table for p↔q

p	q	p↔q
T	T	T
T	F	F
F	T	F
F	F	T

$$\boldsymbol{p \leftrightarrow q \equiv (p \rightarrow q) \wedge (q \rightarrow p)}$$

Logical Equivalence involving $\rightarrow, \leftrightarrow$

- Are the statements of the forms $p \leftrightarrow q$ and $(p \rightarrow q) \wedge (q \rightarrow p)$ logically equivalent?

p	q	p→q	q→p	p↔q	$(\mathbf{p} \rightarrow \mathbf{q}) \wedge (\mathbf{q} \rightarrow \mathbf{p})$
T	T	T	T	**T**	**T**
T	F	F	T	**F**	**F**
F	T	T	F	**F**	**F**
F	F	T	T	**T**	**T**

Truth Table

Order of Logical Operators

1. ~ Evaluate negations first.
2. $\vee$, $\wedge$ Evaluate $\vee$ and $\wedge$ second. If both are present, parentheses may be needed.
3. $\rightarrow$, $\leftrightarrow$ Evaluate $\rightarrow$ and $\leftrightarrow$ third. If both are present, parentheses may be needed.

Arguments

- An **argument** is a sequence of statements, and an **argument form** is a sequence of statement forms. All statements in an argument and all statement forms in an argument form, except for the final one, are called **premises**. The final statement or statement form is called the **conclusion.**
- The symbol ∴, which is read "therefore," is normally placed just before the conclusion.
- To say that an **argument** *form* is **valid** means, if the resulting premises are all true (AND), then the conclusion is also true.

Determining Validity or Invalidity

- An argument form consisting of two premises and a conclusion is called **syllogism**.

If p then q

p

$\therefore q$

$(p \rightarrow q) \wedge p$

premises

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

Critical row

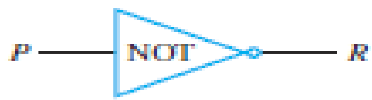
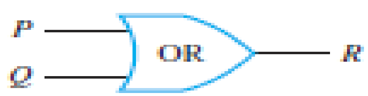
conclusion

It is a valid argument!

Rules of Inference

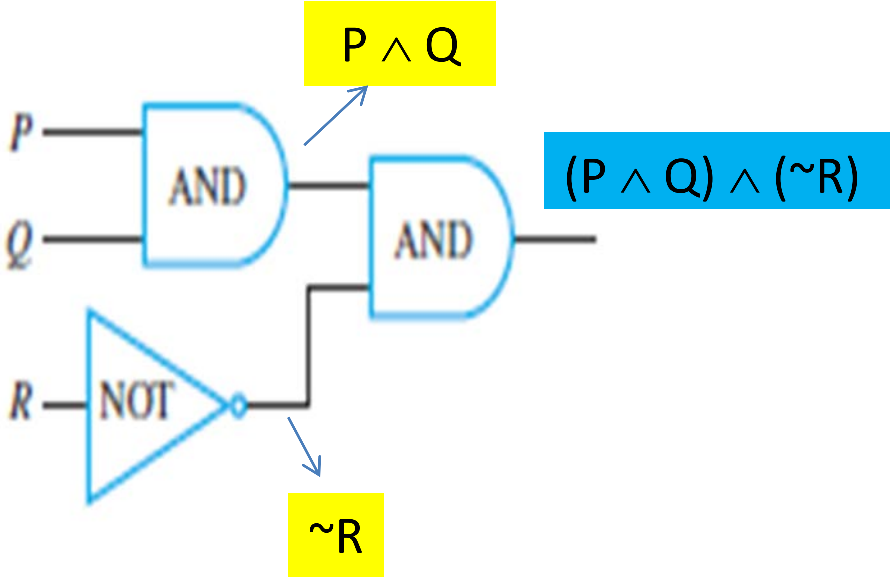
Modus Ponens	$p \to q$ p $\therefore q$	**Elimination**	a. $p \vee q$ $\sim q$ $\therefore p$ b. $p \vee q$ $\sim p$ $\therefore q$
Modus Tollens	$p \to q$ $\sim q$ $\therefore \sim p$	**Transitivity**	$p \to q$ $q \to r$ $\therefore p \to r$
Generalization	a. p $\therefore p \vee q$ b. q $\therefore p \vee q$	**Proof by Division into Cases**	$p \vee q$ $p \to r$ $q \to r$ $\therefore r$
Specialization	a. $p \wedge q$ $\therefore p$ b. $p \wedge q$ $\therefore q$		
Conjunction	p q $\therefore p \wedge q$	**Contradiction Rule**	$\sim p \to c$ $\therefore p$

Digital Logic Circuits

<table>
<tr><th>Type of Gate</th><th>Symbolic Representation</th><th>Action</th></tr>
<tr><td>NOT</td><td>
</td><td>
<table>
<tr><th>Input</th><th>Output</th></tr>
<tr><th>P</th><th>R</th></tr>
<tr><td>1</td><td>0</td></tr>
<tr><td>0</td><td>1</td></tr>
</table>
</td></tr>
<tr><td>AND</td><td>
</td><td>
<table>
<tr><th colspan="2">Input</th><th>Output</th></tr>
<tr><th>P</th><th>Q</th><th>R</th></tr>
<tr><td>1</td><td>1</td><td>1</td></tr>
<tr><td>1</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>1</td><td>0</td></tr>
<tr><td>0</td><td>0</td><td>0</td></tr>
</table>
</td></tr>
<tr><td>OR</td><td>
</td><td>
<table>
<tr><th colspan="2">Input</th><th>Output</th></tr>
<tr><th>P</th><th>Q</th><th>R</th></tr>
<tr><td>1</td><td>1</td><td>1</td></tr>
<tr><td>1</td><td>0</td><td>1</td></tr>
<tr><td>0</td><td>1</td><td>1</td></tr>
<tr><td>0</td><td>0</td><td>0</td></tr>
</table>
</td></tr>
</table>

Boolean Expression for a Circuit

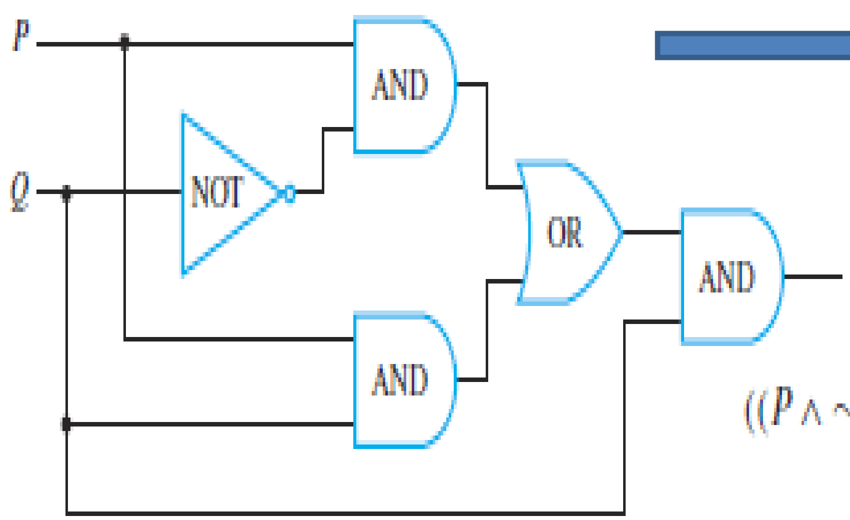
A statement variable or an input signal, that can take one of only two values is called a **Boolean variable.**



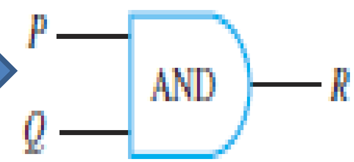
Input			Output
P	Q	R	V
1	1	1	0
1	1	0	1
1	0	1	0
1	0	0	0
0	1	1	0
0	1	0	0
0	0	1	0
0	0	0	0

Circuit for Boolean Expression

$(P \wedge \sim Q) \vee (P \wedge Q) \wedge Q$



Input		Output
P	Q	R
1	1	1
1	0	0
0	1	0
0	0	0



$((P \wedge \sim Q) \vee (P \wedge Q)) \wedge Q$ and $P \wedge Q$, respectively. By Theorem 2.1.1,

$$((P \wedge \sim Q) \vee (P \wedge Q)) \wedge Q$$

$$\equiv (P \wedge (\sim Q \vee Q)) \wedge Q \quad \text{by the distributive law}$$

$$\equiv (P \wedge (Q \vee \sim Q)) \wedge Q \quad \text{by the commutative law for } \vee$$

$$\equiv (P \wedge \mathbf{t}) \wedge Q \quad \text{by the negation law}$$

$$\equiv P \wedge Q \quad \text{by the identity law.}$$

Showing that two circuits are equivalent