Logic

Important for 1. understanding mathematical reasoning
2. design of computer circuits
3. construction of computer programs
4. verification of correctness of programs

Def. A proposition is a statement which is true or false (but not both)

Letters are used to denote propositions: \( \rho, \sigma, \tau, \theta, \ldots \)

Def. Truth value of a proposition is either T (true) or F (false).

Def. Compound propositions are formed from existing propositions using logical operators

Def. Negation of \( \rho \), \( \overline{\rho} \) (other texts use \( \neg \rho \))

A truth table displays the function values of operators

\[ T : \{ T, F \} \rightarrow \{ T, F \} \]

Truth table:

<table>
<thead>
<tr>
<th>( \rho )</th>
<th>( \overline{\rho} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
</tr>
</tbody>
</table>

Def. Conjunction of \( \rho \) and \( \sigma \), \( \rho \land \sigma \), \( \rho \) "and" \( \sigma \)

\[ \land : \{ T, F \} \times \{ T, F \} \rightarrow \{ T, F \} \]

<table>
<thead>
<tr>
<th>( \rho )</th>
<th>( \sigma )</th>
<th>( \rho \land \sigma )</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
</tbody>
</table>

Def. Disjunction of \( \rho \) and \( \sigma \), \( \rho \lor \sigma \), \( \rho \) "or" \( \sigma \)

\[ \lor : \{ T, F \} \times \{ T, F \} \rightarrow \{ T, F \} \] "Inclusive or"

<table>
<thead>
<tr>
<th>( \rho )</th>
<th>( \sigma )</th>
<th>( \rho \lor \sigma )</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
</tbody>
</table>
Def: The exclusive or of \( p \) and \( q \), \( p \oplus q \)
\[ \oplus : \{T,F\} \times \{T,F\} \rightarrow \{T,F\} \]

<table>
<thead>
<tr>
<th>( p )</th>
<th>( q )</th>
<th>( p \oplus q )</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
</tbody>
</table>

Def: The implication of \( p \) and \( q \), denoted \( p \rightarrow q \)
\[ \rightarrow : \{T,F\} \times \{T,F\} \rightarrow \{T,F\} \]

\[ \begin{array}{c|c|c}
 p & q & p \rightarrow q \\
 T & T & T \\
 T & F & F \\
 F & T & T \\
 F & F & T \\
\end{array} \]

if \( p \) then \( q \)
\( p \) implies \( q \)
if \( p \), \( q \)
\( p \) only if \( q \)
\( q \) if \( p \)
\( q \) is sufficient for \( p \)
\( q \) is necessary for \( p \)
\( q \) whenever \( p \)

Note: If...THEN... in computer programming is different from
If...then... in logic.

If \( p \) then \( s \)
if \( p \) is true, execute \( s \)
if \( p \) is false, do not execute \( s \)

Def: The converse of \( p \rightarrow q \) is \( q \rightarrow p \)
The contrapositive of \( p \rightarrow q \) is \( \neg q \rightarrow \neg p \)
The inverse of \( p \rightarrow q \) is \( \neg p \rightarrow \neg q \)

<table>
<thead>
<tr>
<th>( p )</th>
<th>( q )</th>
<th>( p \rightarrow q )</th>
<th>( q \rightarrow p )</th>
<th>( \neg q \rightarrow \neg p )</th>
<th>( \neg p \rightarrow \neg q )</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
<td>T</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
</tbody>
</table>

\[ p \rightarrow q \equiv \neg q \rightarrow \neg p \]
\[ q \rightarrow p \equiv \neg p \rightarrow \neg q \]
Def the biconditional of $p$ and $q$, $p \iff q$, $p$ "iff" $q$,
\[ \iff : \{T,F\} \times \{T,F\} \rightarrow \{T,F\} \]
\[
\begin{array}{c|c|c}
 p & q & p \iff q \\
\hline
 T & T & T \\
 T & F & F \\
 F & T & F \\
 F & F & T \\
\end{array}
\]

Def Bit - one or zero
Allow $F = 0$ $T = 1$

Def Bit operators OR, AND, XOR
 correspond to V, $\land$, $\oplus$

Ex: 

\[
\begin{array}{cccccccc}
0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 \\
1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\
1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \\
\end{array}
\]

\[
\begin{array}{cccccccc}
1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 \\
0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\
1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \\
\end{array}
\]

bitwise OR

bitwise AND

bitwise XOR

Def. A tautology is a compound proposition which
is always true: $p \lor p$
A Contradiction is a compound proposition which
is always false: $p \land \neg p$
A contingency is a compound proposition which
is neither a tautology nor a contradiction

* Compound propositions which have the same truth table are logically equivalent.
To prove $p \iff q$, show $p$ and $q$ have same truth table.
Theorem: Logical Equivalences.

De Morgan's Laws:

\[ 7(p \lor q) \iff 7p \land 7q \]
\[ 7(p \land q) \iff 7p \lor 7q \]
\[ 7p \lor 7q \iff p \to q \]

Distributive Laws:

\[ p \lor (q \land r) \iff (p \lor q) \land (p \lor r) \]
\[ p \land (q \lor r) \iff (p \land q) \lor (p \land r) \]

Proof:

<table>
<thead>
<tr>
<th>p</th>
<th>q</th>
<th>( q \lor r )</th>
<th>( p \lor (q \lor r) )</th>
<th>( p \lor q )</th>
<th>( p \lor r )</th>
<th>( (p \lor q) \land (p \lor r) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
</tbody>
</table>

LHS = RHS

Note: A compound proposition with \( n \) propositions will have \( 2^n \) rows in its truth table.
Rules of Inference

Draw a conclusion or derive a new assertion from old assertions.

**Def** An argument is a set of premises followed by a conclusion: \[ p_1, p_2, \ldots, p_n \rightarrow C \]

Argument is valid or invalid.

**Def** An argument is valid if whenever the premises are true then the conclusion is true.

An argument is invalid if the conclusion can be false when all the premises are true.

**Thm.** To test the validity of an argument, construct the truth table \((p_1 \land p_2 \land \ldots \land p_n) \rightarrow C\).

If it is a tautology, the argument is valid.

If it is not a tautology, the argument is invalid.

Note: Validity of the argument should not be confused with the truth or falsity of the premises and/or the conclusion.

Arguments:

1. \[ p \rightarrow q \]
   \[ \begin{array}{c|c|c}
   p & q & (p \rightarrow q) \rightarrow q \\
   \hline
   T & T & T \\
   T & F & T \\
   F & T & F \\
   F & F & T \\
   \end{array} \]

   Valid.

2. \[ p \rightarrow q \]
   \[ \begin{array}{c|c|c}
   q & p & (p \rightarrow q) \rightarrow p \\
   \hline
   T & T & F \\
   T & F & T \\
   F & T & T \\
   F & F & T \\
   \end{array} \]

   Invalid

3. \[ p \rightarrow q \]
   \[ \begin{array}{c|c|c|c|c|c}
   q & p & (p \rightarrow q) \rightarrow q & T & T & T \\
   \end{array} \]

   Valid
Ex. Prove valid or invalid:

1. If the client was guilty, then he was at the scene of the crime. The client was not at the scene of the crime. Hence, the client was not guilty:
   \[ p \rightarrow q \]
   \[ q \rightarrow r \]
   \[ \therefore \ p \rightarrow r \]

<table>
<thead>
<tr>
<th>P</th>
<th>Q</th>
<th>R</th>
<th>(P \rightarrow Q) \land (Q \rightarrow R) \rightarrow P \rightarrow R</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
<td>T</td>
</tr>
</tbody>
</table>

   Valid.

2. If David passes the final exam, then he will pass the course. David passes the course. Hence, he passed the final exam.
   \[ p \rightarrow q \]
   \[ q \rightarrow p \]
   \[ \therefore \ q \rightarrow p \]

<table>
<thead>
<tr>
<th>P</th>
<th>Q</th>
<th>T</th>
<th>(P \rightarrow Q) \land (Q \rightarrow P) \rightarrow T</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
<td>T</td>
</tr>
</tbody>
</table>

   Valid.

3. The governor will call a special session if the Senate cannot reach a compromise. If a majority of the Cabinet are in agreement then the governor will not call a special session. Therefore, the governor will not call a special session.
   \[ p = \text{Senate cannot reach a compromise} \]
   \[ q = \text{Governor call special session} \]
   \[ r = \text{Majority cabinet in agreement} \]
   \[ T \rightarrow q \]
   \[ p \rightarrow r \]
   \[ \therefore \ q \rightarrow r \]
$\begin{array}{c c c c c c c}
0 & 1 & 2 & 3 & 4 & 5 & 6 \\
T & T & F & T & T & F & T \\
T & T & F & F & T & T & T \\
T & F & F & F & T & T & T \\
T & T & T & T & T & T & T \\
F & F & F & T & T & T & T \\
T & T & T & T & T & T & T \\
\hline
\end{array}$

Invalid

**Shortcut:**

$\begin{align*}
0 & : 7r \to (s \to 7t) \\
1 & : 7r \lor w \\
2 & : 7p \to s \\
3 & : 7w \\
\therefore & : t \to p
\end{align*}$

$\begin{array}{c|c|c|c|c|c|c}
\ p & \ r & \ s & \ t & \ w & [012345678] \to 5 \\
\hline
F & F & T & T & T & T & F \\
\end{array}$

**Argument Valid unless conclusion = F**

$t \to p = F, t = T, p = F$

**Argument Valid unless premises = T**

$\neg w = T, w = F$

$7r \lor w = T, w = F, 7r = T \land r = F$

$7p \to s = T, p = F, T \to s = T \lor s = T$

But $p = F, r = F, s = T, t = T, w = F$

makes $7r \to (s \to 7t) = T \to (T \to F) = T \to F = F$

No way to make premises true and conclusion false, so argument is valid

**Direct Method of Proof** (Formal proof for above)

$\begin{align*}
7r & \to (s \to 7t) \\
7r \lor w & \\
7p & \to s \\
7w & \\
7r & \\
s & \to 7t & \\
7p & \to 7t & \\
t & \to p & \\
\begin{align*}
(7r \lor w) & \land 7w \tag{1} \\
(7r \to (s \to 7t)) & \land 7r \tag{2} \\
(7p \to s) & \land (s \to 7t) \tag{3} \\
converse & \text{positive} & \\
\therefore & p \to r & \\
\end{align*}
\end{align*}$
Propositional Functions and Quantifiers

Def. A propositional function, \( P(x) \) is true or false for each \( x \in U \).

e.g. \( U = \mathbb{Z} \) \( P(x) \): "\( x > 3 \)" \( P(4) = T \), \( P(2) = F \)

In general \( P(x_1, x_2, \ldots, x_n) \) is a propositional function of \( n \) variables.

Def. The universal quantification of \( P(x) \):
"\( P(x) \) is true for all \( x \) in the universe"
\( \forall x \ P(x) \) (for every \( x \))

Ex. Let \( U = \mathbb{Z} \) let \( P(x) \): "\( x + 1 > x \)"
\( \forall x \ P(x) = T \)

let \( Q(x) \): "\( x > 2 \)"
\( \forall x \ Q(x) = F \)

Def. The existential quantification of \( P(x) \) is
"there exists an element \( x \) such that \( P(x) \) is true"
\( \exists x \ P(x) \) (there is an \( x \), there is at least one \( x \), for some \( x \))

Ex. Let \( U = \mathbb{Z} \) let \( P(x) \): "\( x > 3 \)"
\( \exists x \ P(x) = T \)

let \( Q(x) \): "\( x + 1 = x \)"
\( \exists x \ Q(x) = F \)

Commutative law of Addition: \( U = \text{Reals} \)
\( \forall x \forall y \ x + y = y + x \)

Additive Inverse law: \( U = \text{Reals} \)
\( \forall x \exists y \ x + y = 0 \)

Associative law of Addition: \( U = \text{Reals} \)
\( \forall x \forall y \forall z \ x + (y + z) = (x + y) + z \)
Translating Sentences into Logical Expressions

Ex. "Some student in this class has visited Mexico"
\[ U = \text{students in class} \]
\[ M(x): x \text{ has visited Mexico} \]
\[ \exists x \ M(x) \]

Ex. "Every student in this class has visited either Mexico or Canada"
\[ C(x): x \text{ has visited Canada.} \]
\[ \forall x \ (M(x) \lor C(x)) \]

<table>
<thead>
<tr>
<th>Negations</th>
<th>Equivalent Statement</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \neg \exists x \ P(x) )</td>
<td>( \forall x \neg P(x) ) \quad (P(x) \text{ false for every } x)</td>
</tr>
<tr>
<td>( \neg \forall x \ P(x) )</td>
<td>( \exists x \neg P(x) ) \quad (there is one } x \text{ for which } P(x) \text{ is false}</td>
</tr>
</tbody>
</table>

Ex. \( \forall x \exists y \ (\exists x \ A(x,y) \lor \forall x \ B(x,y)) \]

\[ = \forall y \ (\forall x \neg A(x,y) \lor \exists x \ B(x,y)) \]

Ex. Negate the following sentences. Put negation next to the propositional function:

(1) "No one has lost more than $1000 playing the lottery"
\[ L(x) = x \text{ has lost more than $1000 playing the lottery} \]
\[ \forall x \neg L(x) \]

To negate:
\[ \neg \forall x \neg L(x) = \exists x \neg L(x) = \exists x \ L(x) \]
"Someone has lost more than $1000 playing the lottery."
There is a student in class who has passed the final and received a grade of C.

\[ P(x): x \text{ passed final} \quad C(x): x \text{ received grade of C} \]

\[ \exists x \ (P(x) \land C(x)) \]

To negate: \[ \neg (\exists x \ (P(x) \land C(x))) = \forall x \ (\neg P(x) \lor \neg C(x)) \]

"All students in class have not passed the final or not received a grade of C."