Logic

Logic is a discipline that studies the principles and methods used in correct reasoning.

It includes:
- A formal language for expressing statements.
- An inference mechanism (a collection of rules) to reason about valid arguments.
Logic

Logic is a discipline that studies the principles and methods used to construct valid arguments.

An argument is a related sequence of statements to demonstrate the truth of an assertion
- premises are assumed to be true
- conclusion, the last statement of the sequence, is taken to be true based on the truth of the other statements.

An argument is valid if the conclusion follows logically from the truth of the premises.

Logic is the foundation for expressing formal proofs.
Propositional Logic

Propositional Logic is the logic of compound statements built from simpler statements using Logical Boolean connectives.

Some applications

• Design of digital electronic circuits.
• Expressing conditions in programs.
• Queries to databases & search engines.
Propositional Logic

A proposition is a declarative sentence that is either TRUE or FALSE (not both).

Examples:

• The Earth is flat
• $3 + 2 = 5$
• I am older than my mother
• Tallahassee is the capital of Florida
• $5 + 3 = 9$
• Athens is the capital of Georgia
Propositional Logic

- Letters are used to denote propositions.
  - The most frequently used letters are p, q, r, s and t.
- Example:
  
  \[ p: \text{Grass is green.} \]

- The truth value of a proposition is true, denoted by \( T \), if it is true, and false, denoted by \( F \), if it is false.
Propositional Logic

Compound propositions: built up from simpler propositions using logical operators
- Frequently corresponds with compound English sentences.

Example:

Given
p: Jack is older than Jill
q: Jill is female

We can build up
r: Jack is older than Jill and Jill is female \((p \land q)\)
s: Jack is older than Jill or Jill is female \((p \lor q)\)
t: Jack is older than Jill and it is not the case that Jill is female
\[(p \land \neg q)\]
Propositional Logic - negation

Let $p$ be a proposition. The *negation* of $p$ is written $\neg p$ and has meaning:

"It is not the case that $p$."

Truth table for negation:

<table>
<thead>
<tr>
<th>$p$</th>
<th>$\neg p$</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
</tr>
</tbody>
</table>
Propositional Logic - conjunction

**Conjunction operator “∧” (AND):**
- corresponds to English “and.”
- is a binary operator in that it operates on two propositions when creating a compound proposition.

**Def.** Let $p$ and $q$ be two arbitrary propositions, the conjunction of $p$ and $q$, denoted $p \land q$,

$$p \land q,$$

is true if both $p$ and $q$ are true and false otherwise.
Propositional Logic - conjunction

Conjunction operator

\[ p \land q \text{ is true when } p \text{ and } q \text{ are both true.} \]

Truth table for conjunction:

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></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>
Propositional Logic - disjunction

Disjunction operator $\lor$ (or):
- loosely corresponds to English “or.”
- binary operator

**Def.**: Let $p$ and $q$ be two arbitrary propositions, the disjunction of $p$ and $q$, denoted $p \lor q$

is false when both $p$ and $q$ are false and true otherwise.

$\lor$ is also called *inclusive or*

- Observe that $p \lor q$ is true when $p$ is true, or $q$ is true, or both $p$ and $q$ are true.
Propositional Logic - disjunction

Disjunction operator

\[ p \lor q \] is true when \( p \) or \( q \) (or both) is true.

Truth table for conjunction:

<table>
<thead>
<tr>
<th></th>
<th></th>
<th>( p \lor q )</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>
Propositional Logic - XOR

Exclusive Or operator ($\oplus$):

- corresponds to English “either...or...” (exclusive form of or)
- binary operator

Def.: Let $p$ and $q$ be two arbitrary propositions, the exclusive or of $p$ and $q$, denoted $p \oplus q$, is true when either $p$ or $q$ (but not both) is true.
Propositional Logic - XOR

Exclusive Or:

\( p \oplus q \) is true when \( p \) or \( q \) (not both) is true.

Truth table for exclusive or:

<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>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>
Propositional Logic - Implication

Implication operator ($\rightarrow$):

- binary operator
- similar to the English usage of “if...then...”, “implies”, and many other English phrases

Def.: Let $p$ and $q$ be two arbitrary propositions, the implication $p \rightarrow q$ is false when $p$ is false and $q$ is true, and true otherwise.

$$p \rightarrow q \text{ is true when } p \text{ is true and } q \text{ is true, or } p \text{ is false.}$$

$$p \rightarrow q \text{ is false when } p \text{ is true and } q \text{ is false.}$$

Example:

$r$: "The dog is barking."
$s$: "The dog is awake."

$r \rightarrow s$: "If the dog is barking then the dog is awake."
Propositional Logic - Implication

Truth table for implication:

<table>
<thead>
<tr>
<th>p</th>
<th>q</th>
<th>p → q</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>T</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
</tr>
</tbody>
</table>

❖ If the temperature is below 10° F, then water freezes.
Propositional Logic- Biconditional

Biconditional operator (↔):

- Binary operator
- Partly similar to the English usage of “If and only if

Def.: Let $p$ and $q$ be two arbitrary propositions. The \textit{biconditional} $p \leftrightarrow q$ is true when $q$ and $p$ have the same truth values and false otherwise.

Example:

$p$: “The dog plays fetch.”
$q$: “The dog is outside.”

$p \leftrightarrow q$: “The plays fetch if and only if it is outside.”
Propositional Logic - Biconditional

Truth table for biconditional:

<table>
<thead>
<tr>
<th>p</th>
<th>q</th>
<th>p \leftrightarrow q</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>T</td>
</tr>
</tbody>
</table>
Propositional Equivalences

A **tautology** is a proposition that is always true.
- Ex.: \( p \lor \neg p \)

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

A **contradiction** is a proposition that is always false.
- Ex.: \( p \land \neg p \)

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

A **contingency** is a proposition that is neither a tautology nor a contradiction.
- Ex.: \( p \rightarrow \neg p \)

<table>
<thead>
<tr>
<th>( p )</th>
<th>( \neg p )</th>
<th>( p \rightarrow \neg p )</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
</tbody>
</table>
Propositional Logic: Logical Equivalence

If \( p \) and \( q \) are propositions, then \( p \) is logically equivalent to \( q \) if their truth tables are the same.

- “\( p \) is equivalent to \( q \).” is denoted by \( p \equiv q \)

\( p, q \) are logically equivalent if their biconditional \( p \leftrightarrow q \) is a tautology.
Propositional Logic: Logical Equivalence

\[ p \equiv \neg \neg p \]

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

The equivalence holds since these two columns are the same.
Propositional Logic: Logical Equivalence

\[ p \rightarrow q \equiv \neg p \lor q \ ? \]

\[ \text{Does } (p \rightarrow q) \leftrightarrow (\neg p \lor q) \ ? \]

\[ \text{Does } (p \rightarrow q) \rightarrow (\neg p \lor q) \text{ and } (\neg p \lor q) \rightarrow (p \rightarrow q) \ ? \]
Propositional Logic: Logical Equivalences

- **Identity**
  \[ p \land T \equiv p \]
  \[ p \lor F \equiv p \]

- **Domination**
  \[ p \lor T \equiv T \]
  \[ p \land F \equiv F \]

- **Idempotent**
  \[ p \lor p \equiv p \]
  \[ p \land p \equiv p \]

- **Double negation**
  \[ \neg(\neg p) \equiv p \]
Propositional Logic: Logical Equivalences

- **Commutative:**
  
  \[ p \lor q \equiv q \lor p \]
  \[ p \land q \equiv q \land p \]

- **Associative:**
  
  \[ (p \lor q) \lor r \equiv p \lor (q \lor r) \]
  \[ (p \land q) \land r \equiv p \land (q \land r) \]
Propositional Logic: Logical Equivalences

- **Distributive:**
  
  \[ p \lor (q \land r) \equiv (p \lor q) \land (p \lor r) \]
  
  \[ p \land (q \lor r) \equiv (p \land q) \lor (p \land r) \]

- **De Morgan’s:**
  
  \[ \neg (p \land q) \equiv \neg p \lor \neg q \quad \text{(De Morgan’s I)} \]
  
  \[ \neg (p \lor q) \equiv \neg p \land \neg q \quad \text{(De Morgan’s II)} \]
Propositional Logic: Logical Equivalences

- **Absorption:**
  \[ p \lor (p \land q) \equiv p \]
  \[ p \land (p \lor q) \equiv p \]

- **Negation:**
  \[ p \land \neg p \equiv \text{F} \]
  \[ p \lor \neg p \equiv \text{T} \]

- **A useful LE involving →:**
  \[ p \rightarrow q \equiv \neg p \lor q \]
Define:

\[ \text{UGA}(x) = \text{“x is a UGA student.”} \]

**Universe of Discourse** – *all people*

\( x \) is a variable that represents an arbitrary individual in the Universe of Discourse

A **predicate** \( P \), or propositional function, is a function that maps objects in the universe of discourse to propositions

- \( \text{UGA(Paris Hilton)} \) is a **proposition**.
- \( \text{UGA}(x) \) is **not a proposition**.

\( \text{UGA}(x) \) is like an English predicate template

- \( \text{__________} \) is a UGA student
Predicate Logic

\[ \text{Pos}(x) : x > 0 \]

**Universe of Discourse (UoD): Integers**

- \[ \text{Pos}(1) : 1 > 0 = T \]
- \[ \text{Pos}(50) : 50 > 0 = T \]
- \[ \text{Pos}(-10) : -10 > 0 = F \]

\[ \text{Female}(x) : x \text{ is female} \]

**UoD: all people**

- \[ \text{Female}(\text{Maria}) = T \]
- \[ \text{Female}(\text{Edward}) = F \]
Predicate Logic

A predicate that states a property about one object is called a monadic predicate.

- UGA-Student(x)
- Parent(x)
- Female(x)
A predicate of the form $P(x_1, \ldots, x_n)$, $n > 1$ that states the relationships among the objects $x_1, \ldots, x_n$ is called polyadic.

Also, an $n$-place predicate or $n$-ary predicate (a predicate with arity $n$).

- It takes $n > 1$ arguments
- $UoD(x_1, \ldots, x_n) = UoD(x_1) \times UoD(x_2) \times \ldots \times UoD(x_n)$

$L(x, y)$: $x$ loves $y$

$UoD(x) = UoD(y) = \text{all people}$

$L(\text{Adam}, \text{Mary}) = \text{T} \quad L(\text{Mike}, \text{Jill}) = \text{F}$

$Q(x, y, z)$: $x + y = z$

$UoD(x) = UoD(y) = UoD(z): \text{positive integers}$

$Q(1, 1, 2) = \text{T} \quad Q(1, 2, 5) = \text{F}$
Predicate Logic: Universal Quantifier

Suppose that \( P(x) \) is a predicate on some universe of discourse.

The universal quantification of \( P(x) \) (\( \forall x \ P(x) \)) is the proposition:

\( \forall x \ P(x) \) reads “for all \( x \), \( P(x) \) is True”

\( \forall x \ P(x) \) is TRUE means \( P(x) \) is true for all \( x \) in \( \text{UoD}(x) \).

\( \forall x \ P(x) \) is FALSE means there is an \( x \) in \( \text{UoD}(x) \) for which \( P(x) \) is false.
Predicate Logic: Universal Quantifier

- ID(x) means x has a student ID.
  - UoD(x): UGA students
  - \( \forall x \ ID(x) \) means every UGA student has a student ID

- IsFemale(x) means x is a Female
  - UoD(x): UGA students
  - \( \forall x \ IsFemale(x) \) means Every UGA student is a female!

Jim is a UGA student who is a male
Jim is called a **counterexample** for \( \forall x \ IsFemale(x) \)
Predicate Logic: Universal Quantifier

In the special case that the universe of discourse U, is finite, \((U = \{a_1, a_2, a_3, ..., a_n\})\)

\[ \forall x \ P(x) \]

corresponds to the proposition:

\[ P(a_1) \land P(a_2) \land ... \land P(a_n) \]

We can write a program to loop through the elements in the universe and check each for truthfulness.

If all are true, then the proposition is true.

Otherwise it is false!
Predicate Logic: Existential Quantifier

Suppose $P(x)$ is a predicate on some universe of discourse.

The existential quantification of $P(x)$ is the proposition:

“There exists at least one $x$ in the universe of discourse such that $P(x)$ is true.”

$\exists x P(x)$ reads “for some $x$, $P(x)$” or “There exists $x$, $P(x)$ is True”

$\exists x P(x)$ is **TRUE** means

there is an $x$ in UoD(x) for which $P(x)$ is true.

$\exists x P(x)$ is **FALSE** means:

for all $x$ in UoD(x), $P(x)$ is false
Predicate Logic: Existential Quantifier

Examples:

- F(x) means x is female.
  
  UoD(x): UGA students

  ∃ x F(x) means
  
  - For some UGA student x, x is a female.
  - There exists a UGA student x who is a female.

- Y(x) means x is less than 13 years old
  
  UoD(x): UGA students

  ∃ x Y(x) means
  
  - for some UGA student x, x is less than 13 years old
  - there exists a UGA student x such that x is less than 13 years old

What will be the truth-value of ∃ x Y(x) if the UoD is all people?
Predicate Logic: Existential Quantifier

In the special case that the universe of discourse, $U$, is finite, ($U = \{x_1, x_2, x_3, \ldots, x_n\}$)

$$\exists x \ P(x)$$

corresponds to the proposition:

$$P(x_1) \lor P(x_2) \lor \ldots \lor P(x_n)$$

We can write a program to loop through the elements in the universe and check each for truthfulness. If all are false, then the proposition is false. Otherwise, it is true!
Predicates - Quantifier negation

\( \exists x \ P(x) \) means “\( P(x) \) is true for some \( x \).”

What about \( \neg \exists x \ P(x) \)?

It is not the case that [“\( P(x) \) is true for some \( x \).”]

“\( P(x) \) is not true for all \( x \).”

\( \forall x \ \neg P(x) \)

Existential negation:

\( \neg \exists x \ P(x) \equiv \forall x \ \neg P(x). \)
Predicates - Quantifier negation

\( \forall x \ P(x) \) means “\( P(x) \) is true for every \( x \).”

What about \( \neg \forall x \ P(x) \)?

It is not the case that [“\( P(x) \) is true for every \( x \).”]

“There exists an \( x \) for which \( P(x) \) is not true.”

\( \exists x \ \neg P(x) \)

Universal negation:

\( \neg \forall x \ P(x) \equiv \exists x \ \neg P(x). \)
Re-Cap

A **predicate** $P$, or propositional function, is a function that maps objects in the **universe of discourse** to propositions.

- Predicates can be quantified using the universal quantifier ("for all") $\forall$ or the existential quantifier ("there exists") $\exists$.

- Quantified predicates can be negated as follows:
  - $\neg\forall x \ P(x) \equiv \exists x \ \neg P(x)$
  - $\neg\exists x \ P(x) \equiv \forall x \ \neg P(x)$

- Quantified variables are called “bound”.
- Variables that are not quantified are called “free”.
Proofs

A *theorem* is a statement that can be proved to be true.

A *proof* is a sequence of statements that form an argument.
Proofs: Inference Rules

An Inference Rule:

\[ \text{premise 1} \]
\[ \text{premise 2} \ldots \]
\[ \therefore \text{conclusion} \]

“\[ \therefore \]” means “therefore”
Proofs: Modus Ponens

I have a total score over 96.

If I have a total score over 96, then I get an A for the class.

∴ I get an A for this class

\[
p
\quad p \rightarrow q
\quad \quad (p \land (p \rightarrow q)) \rightarrow q
\quad \quad \therefore q
\]

Tautology:
Proofs: Modus Tollens

If the power supply fails then the lights go out.

The lights are on.

∴ The power supply has not failed.

\[ \neg q \]
\[ p \rightarrow q \]
\[ \therefore \neg p \]

Tautology:

\[ \neg q \wedge (p \rightarrow q) \rightarrow \neg p \]
Proofs: Addition

I am a student.

∴ I am a student or I am a visitor.

\[ p \rightarrow (p \lor q) \]

Tautology:

\[ p \rightarrow (p \lor q) \]
Proofs: Simplification

I am a student and I am a soccer player.

∴ I am a student.

\[ p \land q \]

\[ \therefore p \]

Tautology:
\[ (p \land q) \rightarrow p \]
Proofs: Conjunction

I am a student.
I am a soccer player.

∴ I am a student and I am a soccer player.

$p$
$q$

$\therefore p \land q$

Tautology:

$((p) \land (q)) \rightarrow p \land q$
Proofs: Disjunctive Syllogism

I am a student or I am a soccer player.
I am a not soccer player.

∴ I am a student.

\[
p \lor q
\]
\[
\neg q
\]
\[
\therefore p
\]

Tautology:

\[
((p \lor q) \land \neg q) \rightarrow p
\]
Proofs: Hypothetical Syllogism

If I get a total score over 96, I will get an A in the course.

If I get an A in the course, I will have a 4.0 average.

∴ If I get a total score over 96 then I will have a 4.0 average.

\[ p \rightarrow q \]
\[ q \rightarrow r \]
\[ \therefore p \rightarrow r \]

Tautology:
\[ ((p \rightarrow q) \land (q \rightarrow r)) \rightarrow (p \rightarrow r) \]
Proofs: Resolution

I am taking CS4540 or I am taking CS4560.
I am not taking CS4540 or I am taking CS7300.
∴ I am taking CS4560 or I am taking CS7300.

\[ p \lor q \]
\[ \neg p \lor r \]

Tautology:
\[ ((p \lor q) \land (\neg p \lor r)) \rightarrow (q \lor r) \]
Fallacy of Affirming the Conclusion

If you have the flu then you’ll have a sore throat.

You have a sore throat.

∴ You must have the flu.

\[
\begin{align*}
q \\
p \rightarrow q \\
(\therefore p \\
(q \land (p \rightarrow q)) \rightarrow p
\end{align*}
\]

Abductive reasoning (Useful in real life but not in formal predicate logic)
Fallacy of Denying the Hypothesis

If you have the flu then you’ll have a sore throat.

You do not have the flu.

∴ You do not have a sore throat.

\[
\neg p \\
p \to q \\
\therefore \neg q
\]

Fallacy:

\[
(\neg p \land (p \to q)) \to \neg q
\]
Inference Rules for Quantified Statements

\[ \forall x \ P(x) \quad \therefore \ P(c) \]

**Universal Instantiation**
(for an arbitrary object \( c \) from UoD)

\[ P(c) \quad \therefore \ \forall x \ P(x) \]

**Universal Generalization**
(for any arbitrary element \( c \) from UoD)

\[ \exists x \ P(x) \quad \therefore \ P(c) \]

**Existential Instantiation**
(for some specific object \( c \) from UoD)

\[ P(c) \quad \therefore \ \exists x \ P(x) \]

**Existential Generalization**
(for some object \( c \) from UoD)