Artificial Intelligent Systems

(BMA-EL-IZB-LJ-RE 1. year 2024/2025)

Reasoning Systems

Simon Dobrišek

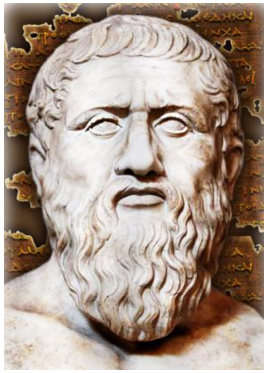
Lecture Topics

- Introduction
- Logical reasoning
- The basics of propositional calculus
- Natural deduction system
- The basics of predicate calculus

INTRODUCTION

- Logic is the branch of philosophy that deals with the use and study of valid reasoning.
- Logic deals with analyzing the patterns of reasoning by which a conclusion is properly drawn from a set of premises.
- Logic can be seen as the relationship and interdependence of a series of events, facts.
- The main types of logic are: informal logic, formal logic, symbolic logic and mathematical logic.

INFORMAL LOGIC



- **Informal** logic is the study of natural language arguments.
- An argument is a series of statements typically used to persuade someone of something or to present reasons for accepting a conclusion.
- An especially important branch of informal logic is the study of fallacies, i.e. the use of invalid or otherwise faulty reasoning for the construction of an argument.
- The dialogues of Plato are good examples of informal logic.

Logical Fallacies

- A logical fallacy is an incorrect argument in logic and rhetoric which undermines an argument's logical validity/soundness.

E.g., Hitler liked dogs. Therefore, dogs are bad.

- Fallacies are either formal fallacies or informal fallacies.
- Informal fallacies are arguments that are fallacious for reasons other than structural (formal) flaws and usually require examination of the argument's content.
- There exists lists of common informal logic fallacies, e.g., https://en.wikipedia.org/wiki/List_of_fallacies

E.g., Birthdays are good for your health. Studies have shown that people who have more birthdays live longer.

Examples of Informal Logical Fallacies

- *Accident* – an exception to a generalization is ignored.

  Cutting people with knives is a crime; Surgeons cut people with knives. → Surgeons are criminals.

- *Appeal to fear* – a specific type of appeal to emotion where an argument is made by increasing fear and prejudice towards the opposing side.

  If you cannot graduate from high school, you will live in poverty for the rest of your life.

- *Appeal to tradition* (*argumentum ad antiquitatem*) – a conclusion supported solely because it has long been held to be true.

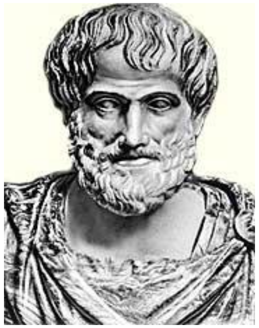
  Drinking boiling hot water (and never cold water) is healthy because Chinese people have done it for thousands of years.

- *Appeal to novelty* (*argumentum novitatis/antiquitatis*) – where a proposal is claimed to be superior or better solely because it is new or modern.

  Upgrading all your software to the most recent versions will make your system more reliable.

- …

Formal Logic



- **Formal** logic is the study of inference with purely formal content.
- The formal logical form of an argument is displayed in the formal grammar of a logical language to make its content usable in formal inference.
- The logical form of a sentence is traditionally defined by
  - a subject (with quantity)
  - a copula (linking the subject with a predicate)
  - a predicate (a subject complement, a property that a subject has or is characterized by).
- The works of Aristotle contain the earliest known formal study of logic and the modern formal logic follows and expands on Aristotle.

Symbolic and Mathematical Logic

- **Symbolic logic** is the study of symbolic abstractions that capture the formal features of logical inference.
- Symbolic logic is divided into two branches: propositional logic and predicate logic.
- **Mathematical logic** is an extension of symbolic logic into other areas, in particular to the study of proof theory, set theory, and recursion theory.
- Agreement on what logic is has remained elusive, and although the field of "***universal logic***" has studied the common structure of logics, there is no widely acceptable formal definition of 'a logic'.

Logical Reasoning

- Reasoning is the act of deriving a conclusion from certain premises using a given methodology.
- Reasoning is a process of thinking, logically arguing, and drawing inference.
- When a system is required to do something, that it has not been explicitly told how to do, it must reason. It must figure out what it can be inferred from what it already knows.
- Human reasoning capabilities are divided into three areas:
  - Mathematical Reasoning – axioms, definitions, theorems, proofs.
  - Logical Reasoning – deductive, inductive, abductive.
  - Non-Logical Reasoning – linguistic , language.

LOGICAL REASONING

- Reasoning skills are often summarized or measured using different quotients.
- The IQ (Intelligence Quotient) is mostly the summation of mathematical reasoning skill and the logical reasoning.
- The EQ (Emotional Quotient) depends mostly on non-logical reasoning capabilities (social skills etc).
- In the field of Artificial intelligence logical reasoning is mostly concerned.

Logical Deduction

- Examples:

> *"If it is rainy, John carries an umbrella.*
> *It is rainy. Therefore, John carries an umbrella."*

> *"If it is rainy, John carries an umbrella. John does not carry an umbrella. Therefore, it is not rainy."*

- Logical deduction is based on applying a general principle to a special case, i.e., using theory to make predictions.
- Reason from facts and general principles to other facts.
- Guarantees that the conclusion is true.
- Usage: Inference engines, Theorem provers, etc.

Logical Induction

- Examples:

  "The grass has been wet every time it has rained. Thus, when it rains, the grass gets wet."

  "People who had cowpox did not get smallpox. Therefore, cowpox prevents smallpox."

- Logical induction is based on deriving a general principle from special cases, i.e., building knowledge from observations and generalizations.
- Reasoning from many instances to all instances.
- Usage: Neural nets, Bayesian nets, Pattern recognition, etc.

Logical Abduction

- Examples:

> **Fact:** *"Mary asks John to a party".*
> **Inferences:** *"Mary likes John."*
> **or** *"John is Mary's last choice."*
> **or** *"Mary wants to make someone else jealous."*

- Logical abduction is based on guessing that some general principle can relate to a given pattern of cases, i.e., extracting hypotheses to form a tentative theory.
- Common form of human reasoning– "Inference to the best explanation".
- Usage: Knowledge discovery, Statistical methods, Data mining.

The Basics of Propositional Calculus

- Propositional calculus is concerned with the study of propositions.
- Propositions are declarative statements/sentences that can be either true or false, and nothing else.
- Propositions are formed by other propositions with the use of logical connectives (logical operators), such as negation, conjunction, disjunction, implications, etc.
- In propositional calculus, statements, or propositions, can be replaced with variable names.

It's raining and I'm wet. $\longleftrightarrow$ $(P \wedge Q)$

Examples of Propositions

“Drilling for oil caused dinosaurs to become extinct.”

“$x + 2 = 2x$ when $x = -2$”

“All cows are brown.”

“Birds have wings.”

“The Earth is further from the sun than Venus.”

“There is life on Mars”

“$3 \times 3 = 8$”

“The square is a rectangle.”

Examples of Sentences/Statements that are not Propositions

"It is cold outside?"
(Since a question is not a declarative sentence, it fails to be a proposition)

"Close the door!"
(Likewise, an imperative is not a declarative sentence; hence, fails to be a proposition)

"The value of x is greater than 2."
(This is a declarative sentence, but unless x is assigned a value or is otherwise prescribed, the sentence is neither true nor false, hence, not a proposition).

"This sentence is false."
(This example is called a paradox and is not a proposition, because it is neither true nor false)

Logical Connectives

- A logical connective (a logical operator) is a symbol or word used to connect two or more propositions, such that the sense of the compound proposition produced depends only on the original propositions.
- The most common logical connectives are binary connectives which join two sentences.
- Negation is considered to be a unary connective.
- Logical connectives along with quantifiers are the two main types of logical constants used in formal systems such as propositional logic and predicate logic.

Logical Connectives

In the grammar of natural languages two sentences may be joined by a grammatical conjunction to form a grammatically compound sentence.

- *"and"* (conjunction)
- *"and then"* (conjunction)
- *"and then within"* (conjunction)
- *"or"* (disjunction)
- *"either...or"* (exclusive disjunction)
- *"implies"* (implication)
- *"if...then"* (implication)
- *"if and only if"* (equivalence)
- *"only if"* (implication)
- *"just in case"* (biconditional)
- *"but"* (conjunction)
- *"however"* (conjunction)
- *"not both"* (alternative denial)
- *"neither...nor"* (joint denial)

Common Logical Connectives

Commonly used logical connectives include

- Negation (not): $\neg$ $\sim$ $\overline{\ }$
- Conjunction (and): $\cdot$ & $\wedge$
- Disjunction (or): $+$ $\vee$
- Material implication (if…then): $\Rightarrow$ $\longrightarrow$ $\supset$
- Biconditional (if and only if, iff,): $\Leftrightarrow$ $\longleftrightarrow$ $\equiv$

Common Logical Connectives

The meaning of the statements *"It is raining"* and *"I am indoors"* is transformed when the two are combined with logical connectives (P = *"It is raining"* and Q = *"I am indoors"*):

- *"It is* **not** *raining"* ($\neg P$)
- *"It is raining* **and** *I am indoors"* ($P \wedge Q$)
- *"It is raining* **or** *I am indoors"* ($P \vee Q$)
- **"If** *it is raining,* **then** *I am indoors"* ($P \rightarrow Q$)
- **"If** *I am indoors,* **then** *it is raining"* ($Q \rightarrow P$)
- *"I am indoors* **if and only if** *it is raining"* ($P \leftrightarrow Q$)

Truth Table

- A truth table is a mathematical table used to compute the functional values of logical expressions on each of their functional arguments.
- Let us consider that a proposition is either true (denoted either T or 1) or false (denoted either F or 0).
- A truth table for the main binary logical connectives are then:

P	Q	$P \vee Q$	$P \wedge Q$	$P \rightarrow Q$	$P \leftrightarrow Q$
T	T	T	T	T	T
T	F	T	F	F	F
F	T	T	F	T	F
F	F	F	F	T	T

Material Implication (If…Then)

- Material implication is often confused with logical implication.
- Material implication is a binary connective that is used to create new sentences; so $P \rightarrow Q$ is a compound sentence using the material implication symbol $\rightarrow$.
- Logical implication (inference) is a relation between two sentences P and Q, which says that any model that makes P true also makes Q true.
- This can be written as $P \vDash Q$ or $P \vdash Q$, or sometimes, confusingly, as $P \Longrightarrow Q$, as some people use $\Longrightarrow$ for material implication as well.

Material Implication Example

P: *"This book is interesting."*
Q: *"I am staying at home."*

$P \rightarrow Q$:
"**If** *this book is interesting*, **then** *I am staying at home.*"

- Equivalent forms of "**If** P **then** Q":
  - "P **implies** Q"
  - "**If** P, Q"
  - "P **is a sufficient condition for** Q"
  - "Q **if** P"
  - "Q **whenever** P"
  - "Q **is a necessary condition for** P"

Material Implication Interpretation

- The meaning of $P \rightarrow Q$ should be seen as a contract that says, if the first condition is satisfied, then the second will also be satisfied.
- If the first condition, P, is not satisfied, then the condition of the contract is null and void. In this case, it does not matter if the second condition is satisfied or not, the contract is still upheld.

Consider the proposition –
"If a body is stationary then the sum of forces on it is zero."

P	Q	$P \rightarrow Q$
T	T	T
T	F	F
F	T	T
F	F	T

Propositional Equivalences

- Propositions P and Q are logically equivalent if $P \leftrightarrow Q$ is a tautology (always true).
- One of the ways to prove that two propositions are equivalent is using a truth table.
- For example, let us prove conditional equivalence $(P \to Q) \leftrightarrow (\neg P \vee Q)$

P	Q	$\neg P$	$\neg P \vee Q$	$P \to Q$
T	T	F	T $\leftrightarrow$	T
T	F	F	F $\leftrightarrow$	F
F	T	T	T $\leftrightarrow$	T
F	F	T	T $\leftrightarrow$	T

PROPOSITIONAL EQUIVALENCES

- Some equivalences are important enough to have names:

Name	Equivalences
Double Negation	$\neg(\neg P) \leftrightarrow P$
Commutative	$P \lor Q \leftrightarrow Q \lor P$ $P \land Q \leftrightarrow Q \land P$
Associative	$(P \lor Q) \lor R \leftrightarrow P \lor (Q \lor R)$ $(P \land Q) \land R \leftrightarrow P \land (Q \land R)$
Distributive	$P \lor (Q \land R) \leftrightarrow (P \lor Q) \land (P \lor R)$ $P \land (Q \lor R) \leftrightarrow (P \land Q) \lor (P \land R)$
De Morgan's Law	$\neg(P \land Q) \leftrightarrow \neg P \lor \neg Q$ $\neg(P \lor Q) \leftrightarrow \neg P \land \neg Q$
Absorption	$P \lor (P \land Q) \leftrightarrow P$ $P \land (P \lor Q) \leftrightarrow P$
Negation	$P \lor \neg P \leftrightarrow T$ $P \land \neg P \leftrightarrow F$
…	

Propositional Equivalences

- Equivalences can be proved also mathematically using the existing ones, i.e. without drawing a truth table.
- For example, let us prove that $Q \land \neg(P \rightarrow Q)$ is a contradiction:

$$
\begin{aligned}
Q \land \neg(P \rightarrow Q) \quad &\leftrightarrow \quad Q \land \neg(\neg P \lor Q) && \text{[Conditional equivalence]} \\
&\leftrightarrow \quad Q \land (\neg\neg P \land \neg Q) && \text{[De Morgan's]} \\
&\leftrightarrow \quad Q \land (P \land \neg Q) && \text{[Double Negation]} \\
&\leftrightarrow \quad Q \land (\neg Q \land P) && \text{[Commutative]} \\
&\leftrightarrow \quad (Q \land \neg Q) \land P && \text{[Associative]} \\
&\leftrightarrow \quad F \land P && \text{[Negation]} \\
&\leftrightarrow \quad F && \text{[Domination]}
\end{aligned}
$$

Binary Truth Functions

- There are sixteen possible truth functions of two input propositions P and Q that are usually identified in the propositions expressed in natural languages.
- Any of these functions corresponds to a truth table of a certain logical connective, including several degenerate cases, such as a function not depending on one or both of its arguments.
- Truth and falsehood is denoted as 1 and 0 in the following truth tables, respectively.

Tautologies and Contradictions

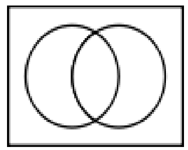
- A proposition that is always true is called a tautology.
- A proposition that is always false is called a contradiction.
- A proposition that is neither a tautology or a contradiction is a contingency (may be true or false).
- Examples:
  - $P \lor \neg P$ is a tautology.
  - $P \land \neg P$ is a contradiction.
  - $\neg P \lor (P \rightarrow Q)$ is which?

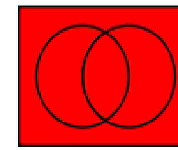
"Crows are black or not black."

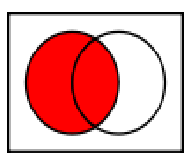
"Mike is married to Mary and Mary is not married to Mike."

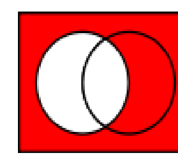
"I will not do it or if I do it then I will fail."

Binary Truth Functions

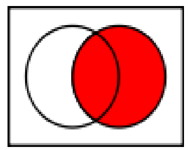
<table>
<tr><th colspan="4">Contradiction/False</th></tr>
<tr><th>Notation</th><th>Equivalent formulas</th><th>Truth table</th><th>Venn diagram</th></tr>
<tr><td>$\bot$
"bottom"</td><td>$P \land \neg P$</td><td><table><tr><td></td><td></td><td colspan="2">Q</td></tr><tr><td></td><td></td><td>0</td><td>1</td></tr><tr><td rowspan="2">P</td><td>0</td><td>0</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr></table></td><td></td></tr>
</table>

<table>
<tr><th colspan="4">Tautology/True</th></tr>
<tr><th>Notation</th><th>Equivalent formulas</th><th>Truth table</th><th>Venn diagram</th></tr>
<tr><td>$\top$
"top"</td><td>$P \lor \neg P$</td><td><table><tr><td></td><td></td><td colspan="2">Q</td></tr><tr><td></td><td></td><td>0</td><td>1</td></tr><tr><td rowspan="2">P</td><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table></td><td></td></tr>
</table>

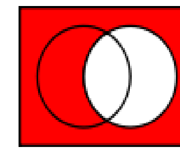
<table>
<tr><th colspan="4">Proposition P</th></tr>
<tr><th>Notation</th><th>Equivalent formulas</th><th>Truth table</th><th>Venn diagram</th></tr>
<tr><td>P</td><td></td><td><table><tr><td></td><td></td><td colspan="2">Q</td></tr><tr><td></td><td></td><td>0</td><td>1</td></tr><tr><td rowspan="2">P</td><td>0</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table></td><td></td></tr>
</table>

<table>
<tr><th colspan="4">Negation of P</th></tr>
<tr><th>Notation</th><th>Equivalent formulas</th><th>Truth table</th><th>Venn diagram</th></tr>
<tr><td>$\neg P$
$\sim P$</td><td></td><td><table><tr><td></td><td></td><td colspan="2">Q</td></tr><tr><td></td><td></td><td>0</td><td>1</td></tr><tr><td rowspan="2">P</td><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>0</td></tr></table></td><td></td></tr>
</table>

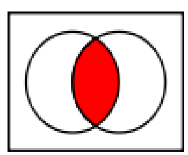
Binary Truth Functions

Proposition Q			
Notation	Equivalent formulas	Truth table	Venn diagram
Q			

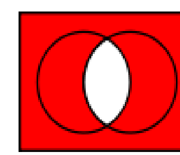
		Q	
		0	1
P	0	0	1
	1	0	1

Negation of Q			
Notation	Equivalent formulas	Truth table	Venn diagram
$\neg Q$ $\sim Q$			

		Q	
		0	1
P	0	1	0
	1	1	0

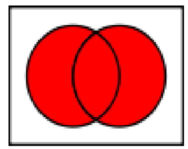
Conjunction			
Notation	Equivalent formulas	Truth table	Venn diagram
$P \wedge Q$ $P \,\&\, Q$ $P \cdot Q$ P AND Q	$P \nrightarrow \neg Q$ $\neg P \nleftarrow Q$ $\neg P \downarrow \neg Q$		

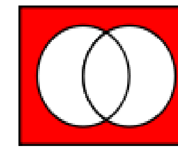
		Q	
		0	1
P	0	0	0
	1	0	1

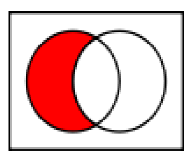
Alternative denial			
Notation	Equivalent formulas	Truth table	Venn diagram
$P \uparrow Q$ $P \mid Q$ P NAND Q	$P \rightarrow \neg Q$ $\neg P \leftarrow Q$ $\neg P \vee \neg Q$		

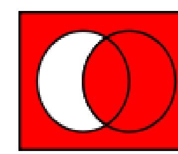
		Q	
		0	1
P	0	1	1
	1	1	0

Binary Truth Functions

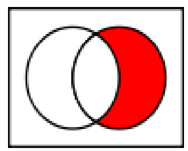
Disjunction			
Notation	Equivalent formulas	Truth table	Venn diagram
$P \lor Q$ P OR Q	$P \leftarrow \neg Q$ $\neg P \rightarrow Q$ $\neg P \uparrow \neg Q$ $\neg(\neg P \land \neg Q)$	Q: 0, 1 P 0: 0, 1 P 1: 1, 1	

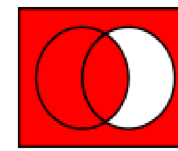
Joint denial			
Notation	Equivalent formulas	Truth table	Venn diagram
$P \downarrow Q$ P NOR Q	$P \nleftarrow \neg Q$ $\neg P \nrightarrow Q$ $\neg P \land \neg Q$	Q: 0, 1 P 0: 1, 0 P 1: 0, 0	

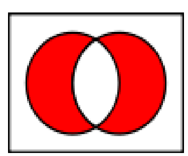
Material nonimplication			
Notation	Equivalent formulas	Truth table	Venn diagram
$P \nrightarrow Q$ $P \not\supset Q$	$P \land \neg Q$ $\neg P \downarrow Q$ $\neg P \nleftarrow \neg Q$	Q: 0, 1 P 0: 0, 0 P 1: 1, 0	

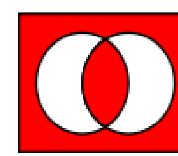
Material implication			
Notation	Equivalent formulas	Truth table	Venn diagram
$P \rightarrow Q$ $P \supset Q$	$P \uparrow \neg Q$ $\neg P \lor Q$ $\neg P \leftarrow \neg Q$	Q: 0, 1 P 0: 1, 1 P 1: 0, 1	

Binary Truth Functions

<table>
<tr><th colspan="4">Converse nonimplication</th></tr>
<tr><th>Notation</th><th>Equivalent formulas</th><th>Truth table</th><th>Venn diagram</th></tr>
<tr>
<td>$P \nleftarrow Q$
$P \not\subset Q$</td>
<td>$P \downarrow \neg Q$
$\neg P \wedge Q$
$\neg P \nrightarrow \neg Q$</td>
<td><table>
<tr><td></td><td></td><td colspan="2">Q</td></tr>
<tr><td></td><td></td><td>0</td><td>1</td></tr>
<tr><td rowspan="2">P</td><td>0</td><td>0</td><td>1</td></tr>
<tr><td>1</td><td>0</td><td>0</td></tr>
</table></td>
<td></td>
</tr>
</table>

<table>
<tr><th colspan="4">Converse implication</th></tr>
<tr><th>Notation</th><th>Equivalent formulas</th><th>Truth table</th><th>Venn diagram</th></tr>
<tr>
<td>$P \leftarrow Q$
$P \subset Q$</td>
<td>$P \vee \neg Q$
$\neg P \uparrow Q$
$\neg P \rightarrow \neg Q$</td>
<td><table>
<tr><td></td><td></td><td colspan="2">Q</td></tr>
<tr><td></td><td></td><td>0</td><td>1</td></tr>
<tr><td rowspan="2">P</td><td>0</td><td>1</td><td>0</td></tr>
<tr><td>1</td><td>1</td><td>1</td></tr>
</table></td>
<td></td>
</tr>
</table>

<table>
<tr><th colspan="4">Exclusive disjunction</th></tr>
<tr><th>Notation</th><th>Equivalent formulas</th><th>Truth table</th><th>Venn diagram</th></tr>
<tr>
<td>$P \nleftrightarrow Q$
$P \not\equiv Q$
$P \oplus Q$
P XOR Q</td>
<td>$P \leftrightarrow \neg Q$
$\neg P \leftrightarrow Q$
$\neg P \nleftrightarrow \neg Q$</td>
<td><table>
<tr><td></td><td></td><td colspan="2">Q</td></tr>
<tr><td></td><td></td><td>0</td><td>1</td></tr>
<tr><td rowspan="2">P</td><td>0</td><td>0</td><td>1</td></tr>
<tr><td>1</td><td>1</td><td>0</td></tr>
</table></td>
<td></td>
</tr>
</table>

<table>
<tr><th colspan="4">Biconditional</th></tr>
<tr><th>Notation</th><th>Equivalent formulas</th><th>Truth table</th><th>Venn diagram</th></tr>
<tr>
<td>$P \leftrightarrow Q$
$P \equiv Q$
P XNOR Q
P IFF Q</td>
<td>$P \nleftrightarrow \neg Q$
$\neg P \nleftrightarrow Q$
$\neg P \leftrightarrow \neg Q$</td>
<td><table>
<tr><td></td><td></td><td colspan="2">Q</td></tr>
<tr><td></td><td></td><td>0</td><td>1</td></tr>
<tr><td rowspan="2">P</td><td>0</td><td>1</td><td>0</td></tr>
<tr><td>1</td><td>0</td><td>1</td></tr>
</table></td>
<td></td>
</tr>
</table>

Natural Deduction System

- Natural deduction is a kind of proof calculus in which logical reasoning is expressed by inference rules closely related to the "natural" way of reasoning.
- Propositional calculus has several inference rules that allow derivation of other true propositions (consequents) given a set of propositions (premises) that are assumed to be true.
- Conjunction Introduction, Modus Ponens and Resolution are the most commonly used inference rules in the field of AI.

Conjunction Introduction

- The Conjunction Introduction rule makes it possible to introduce a conjunction into a logical proof.

- It is the inference that if the proposition P is true, and proposition Q is true, then the logical conjunction of the two propositions $P \wedge Q$ is true.

$$\frac{P, Q}{\therefore P \wedge Q}$$

- The rule can be state as

  Premise 1: P

  Premise 2: Q

  Consequent: $P \wedge Q$

- The rule may be written in sequent notation:

$$P, Q \vdash P \wedge Q$$

where $\vdash$ is a metalogical symbol meaning that $P \wedge Q$ is a syntactic consequence if P and Q are on lines of a proof in the logical system;

Modus Ponens

- Modus ponendo ponens (Latin for "the way that affirms by affirming"; generally abbreviated to MP or modus) or implication elimination is a valid, simple argument form and rule of inference.

- The rule can be state as

$$\frac{P \to Q, P}{\therefore Q}$$

Premise 1: $P \to Q$

Premise 2: P

Consequent: Q

- The rule may be written in sequent notation:

$$P \to Q, P \vdash Q$$

where $\vdash$ is a metalogical symbol meaning that Q is a syntactic consequence of $P \to Q$ and P in some logical system;

- or as the statement of a truth-functional tautology or theorem of propositional logic:

$$((P \to Q) \land P) \to Q$$

Modus Ponens Truth Table

- In instances of modus ponens we assume as premises that $P \rightarrow Q$ is true and P is true.
- If we draw the truth table

Tautology

P	Q	$P \rightarrow Q$	$P \wedge (P \rightarrow Q)$	$(P \wedge (P \rightarrow Q)) \rightarrow \mathrm{Q}$
T	T	T	T	T
T	F	F	F	T
F	T	T	F	T
F	F	T	F	T

- Only the first line of the truth table satisfies these two conditions. On this line, Q is also true. Therefore, whenever $P \rightarrow Q$ is true and P is true, Q must also be true.

Modus Ponens and Forward Chaining

- The first premise is the "if–then" or conditional claim, namely that P implies Q.
- The second premise is that P, the antecedent of the conditional claim, is true.
- From these two premises it can be logically concluded that Q, the consequent of the conditional claim, must be true as well.
- In artificial intelligence, modus ponens is often called forward chaining.
- An example of an argument that fits the form *modus ponens*:

> *"If today is Tuesday, then John will go to work."*
> *"Today is Tuesday."*
> *"Therefore, John will go to work."*

Resolution Inference Rule

- The resolution inference rule is popular in AI, as some logical programming languages, such as Prolog, support only logical connectives $\wedge$,$\vee$ and $\neg$.

$$\frac{P \vee Q, P \vee \neg Q}{\therefore P}$$

- The rule can be state as

Premise 1: $P \vee Q$

Premise 2: $P \vee \neg Q$

Consequent: P

- The rule may be written as the statement of a truth-functional theorem of propositional logic:

$$((P \vee Q) \wedge (P \vee \neg Q)) \rightarrow P$$

Resolution Inference Rule

- The resolution inference rule and logical equivalence can be proved using a truth table.

P	Q	$\neg Q$	$P \vee Q$	$P \vee \neg Q$	$(P \vee Q) \wedge (P \vee \neg Q)$	$((P \vee Q) \wedge (P \vee \neg Q)) \rightarrow \mathrm{P}$
T	T	F	T	T	T	T
T	F	T	T	T	T	T
F	T	F	T	F	F	T
F	F	T	F	T	F	T

Equivalence

Equivalence

Tautology

Other Inference Rules

- Many other inference rules can be derived from the different argument forms.

Name	Sequent	Description
Modus Tollens	$((P \to Q) \land \neg Q) \to \neg P$	If P then Q; not Q; therefore not P
Hypothetical Syllogism	$((P \to Q) \land (Q \to R)) \to (P \to R)$	If P then Q; if Q then R; therefore, if P then R
Constructive Dilemma	$(((P \to Q) \land (R \to S)) \land (P \lor R)) \to (Q \lor S)$	If P then Q; and if R then S; but P or R; therefore Q or S
Destructive Dilemma	$(((P \to Q) \land (R \to S)) \land (\neg Q \lor \neg S)) \to (\neg P \lor \neg R)$	If P then Q; and if R then S; but not Q or not S; therefore not P or not R
…		

The Basics of Predicate Calculus

- The proposition *“X is large than 2”* cannot be considered in the traditional propositional calculus but can be considered in the predicate calculus.
- In the propositional calculus, propositions are considered as a whole, and their internal structure is not further analyzed.
- On the other side, the predicate calculus consider the internal structure of propositions.
- In general, predicate is the part of the sentence (or clause) which states something about the subject and/or the object of the sentence.

> In *"The dog barked very loudly"*, the subject is *"the dog"* and the **predicate** is *"barked very loudly"*.

Predicates in Traditional Grammar

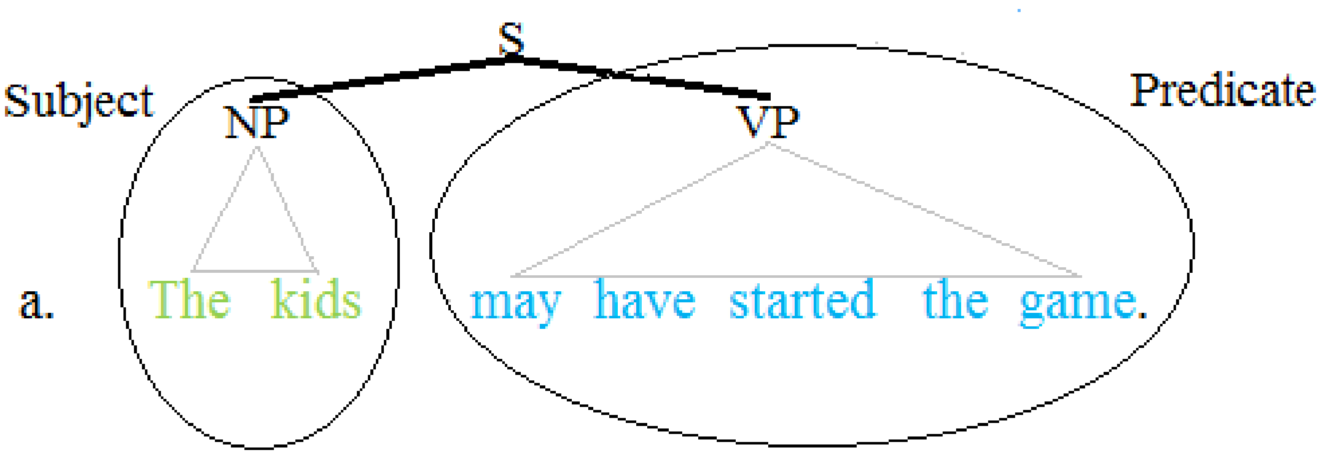
- In the traditional grammar, the predicate (from Late Latin *praedicatum* - “thing said of a subject”) is actually inspired by propositional logic.
- The predicate is one of the two main parts of a sentence (the other being the subject, which the predicate modifies).
- The predicate provides information about the subject, such as what the subject is, what the subject is doing, or what the subject is like.

Examples of Predicates

- *"She **dances**."* - verb-only predicate
- *"Ben **reads the book**."* - verb + direct object predicate
- *"Ben's mother, Felicity, **gave me a present**."* – verb + indirect object + direct object predicate
- *"She **listened to the radio**."* - verb + prepositional object predicate
- *"They **elected him president**."* - verb + object + predicative noun predicate
- *"She **met him in the park**."* - verb + object + adjunct predicate
- *"She **is in the park.**"* – verb + predicative prepositional phrase predicate

Predicates in Traditional Grammar

- In the traditional understanding of predicates, the generic declarative sentence (S) is divided into a noun phrase (NP) and verb phrase (VP), e.g.,



Predicates in modern theories of syntax and grammar

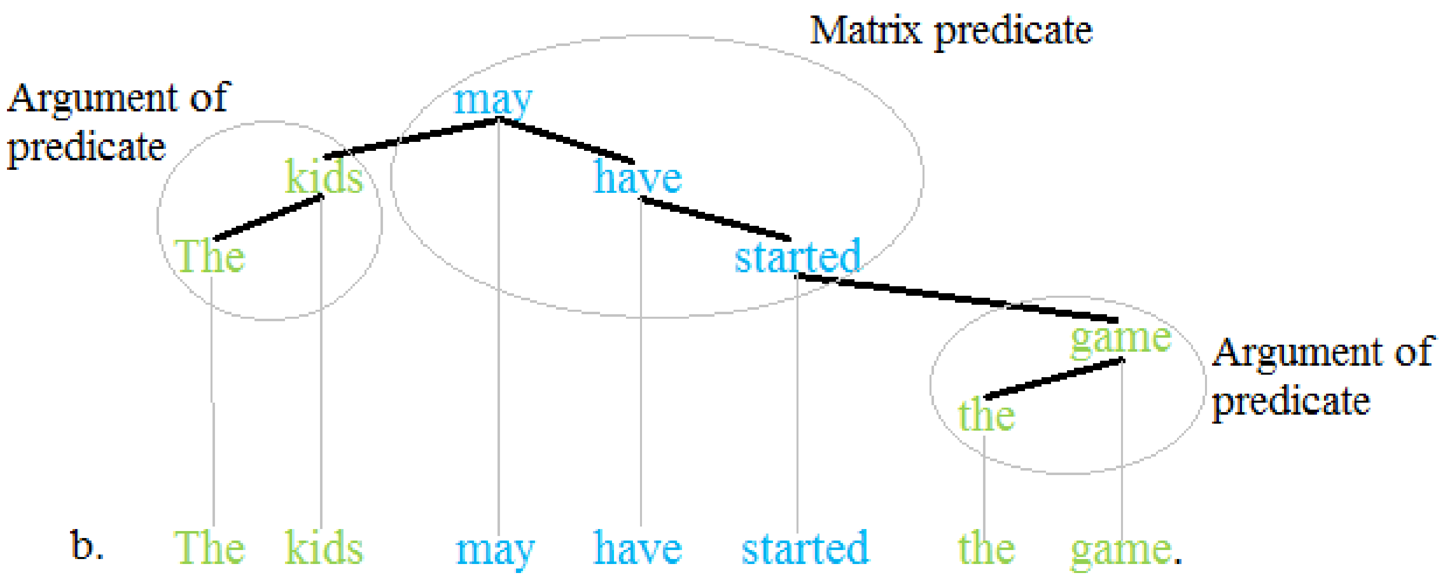
- This understanding sees predicates as relations or functions over arguments.
- The predicate serves either to assign a property to a single argument or to relate two or more arguments to each other.
- Predicates are placed on the left outside of brackets, whereas the predicate's arguments are placed inside the brackets, e.g.,
  - *"Bob **laughed**."* → `laughed('Bob').`
  - *"Sam **helped** you."* → `helped('Sam', 'you').`
  - *"Jim **gave** Jill his dog."* → `gave('Jim', 'Jill', 'his dog').`

Predicates in modern theories of syntax and grammar

- Other function words - e.g. auxiliary verbs, certain prepositions, phrasal particles, etc. - are viewed as part of the predicate.
  - *"Bill **will have laughed**."* → `will_have_laughed('Bill').`
  - ***"Has** that **been funny**?"* → `has_been_funny('that')?`
  - *"You **should give** it **up**."* → `should_give_up('you', 'it').`
  - *"The butter **is in** the drawer."*
    → `is_in('butter', 'drawer').`
  - *"Who **did** Jim **give** his dog **to**?"*
    → `did_give_to('who', 'Jim', 'his dog').`

PREDICATES IN MODERN THEORIES OF SYNTAX AND GRAMMAR

- This modern understanding of predicates is compatible with the dependency grammar approach to sentence structure, which places the finite verb as the root of all structure, e.g.



Predicate Calculus

- The predicate calculus is distinguished from the propositional calculus (and other formal systems) in that its formulae contain variables which can be quantified.
- A predicate variable is a predicate letter which can stand for a relation (between terms) but which has not been specifically assigned any particular relation (or meaning).
- Two common quantifiers are the existential ∃ ("there exists") and universal ∀ ("for all") quantifiers.

Predicate Calculus Syntax

- In the traditional predicate calculus, predicates are written in capital letters (and sometimes shortened) and variables in small letters, e.g.,
  - *"Hans is German."* → `GERMAN(HANS).`
  - *"Peter is in the room."* → `IN(PETER, ROOM).`
  - *"X is grater than Y."* → `GREATER(x, y).`
- The first two predicates have constant arguments and can be considered as true or false propositions.
- The third predicate represents infinite number of prepositions, as the two variables can be assigned to any numbers, and only when the first number is greater than the second one the predicate is considered to be a true proposition.

The Basic Logical Connectives

- The traditional predicate calculus considers the same logical connectives as the propositional calculus, e.g.,
  - *"Peter lives in Paris, France."*

    $\mathrm{LIVES(PETER,PARIS)} \land \mathrm{LIVES(PETER,FRANCE)}.$
  - *"Peter is in the room or in the kitchen."*

    $\mathrm{IN(PETER,ROOM)} \lor \mathrm{IN(PETER,KITCHEN)}.$
  - *"If the owner of the car is Peter then the car is red"*

    $\mathrm{OWN(PETER,CAR)} \rightarrow \mathrm{COLOUR(CAR,RED)}.$
  - *"If Peter does not read the book then he writes the letter."*

    $\overline{\mathrm{READ}}\mathrm{(PETER,BOOK)} \rightarrow \mathrm{WRITE(PETER,LETTER)}.$

Predicate Calculus Syntax

- In the modern predicate calculus, predicates are often written in small letters (and in capital letters if shortened) and variables in small letters as well, e.g.,
  - *“Hans is German.”* → `german('Hans').`
  - *“Peter is in the room.”* → `in('Peter',room).`
  - *“X is greater than Y.”* → `greater(x,y)` or also `GT(x,y).`
- In general, predicates can be seen and written as properties of individual objects or their interrelations:
  - *“Object a has the property P.”* → $P(a)$.
  - *“Objects a_1 and a_2 are in the relation Q”* → $Q(a_1, a_2)$.
  - *“Objects $a_1, a_2, \cdots, a_n$ are in the relation R.”* → $R(a_1, a_2, \cdots, a_n)$.

Predicate Quantifiers

- An existential quantification is a logical constant (∃) that is interpreted as "there exists", "there is at least one", or "for some“.

- Examples:

  “For some natural number n, $n \cdot n = 25$.”

  $(\exists n \in \mathbb{N})P(n, n, 25)$, ← True statement
  where $P(x, y, z)$ is the predicate “*x times y equals z*”.

  “For some natural number n, n is even and $n \cdot n = 25$.”

  $(\exists n \in \mathbb{N})(Q(n) \land P(n, n, 25))$, ← False statement
  where $Q(a)$ is the predicate “*a is even*”.

  “Something green is on the table.”

  $(\exists x)(green(x) \land on(x, 'table'))$. ← Can be true or false

PREDICATE QUANTIFIERS

- A universal quantification is a logical constant ($\forall$) that is interpreted as "given any" or "for all".
- Examples:

  "If x is not equal to y, then x is greater or less than y."

  $(\forall x)(\forall y)(\neg equal(x, y) \rightarrow greater(x, y) \vee less(x, y))$,
  where $equal(a, b)$ is the predicate "*a is equal to b*",
  $greater(a, b)$ is the predicate "*a is greater than b*",
  $less(a, b)$ is the predicate "*a is less than b*".

- If a variable is not coupled to a quantifier then it is free and may be optionally substituted with any constants.

Properties of Predicate Quantifiers

- Many properties of predicate quantifiers have been studied by studying predicate (proposition) equivalences in combination with different logical connectives, e.g.,

$$
\begin{aligned}
&P(x) \wedge (\exists y{\in}\mathbf{Y}\, Q(y)) \equiv \exists y{\in}\mathbf{Y}\,(P(x) \wedge Q(y))\\
&P(x) \vee (\exists y{\in}\mathbf{Y}\, Q(y)) \equiv \exists y{\in}\mathbf{Y}\,(P(x) \vee Q(y)), \text{ provided that } \mathbf{Y} \neq \emptyset\\
&P(x) \rightarrow (\exists y{\in}\mathbf{Y}\, Q(y)) \equiv \exists y{\in}\mathbf{Y}\,(P(x) \rightarrow Q(y)), \text{ provided that } \mathbf{Y} \neq \emptyset\\
&P(x) \nleftarrow (\exists y{\in}\mathbf{Y}\, Q(y)) \equiv \exists y{\in}\mathbf{Y}\,(P(x) \nleftarrow Q(y))\\
&P(x) \wedge (\forall y{\in}\mathbf{Y}\, Q(y)) \equiv \forall y{\in}\mathbf{Y}\,(P(x) \wedge Q(y)), \text{ provided that } \mathbf{Y} \neq \emptyset\\
&P(x) \vee (\forall y{\in}\mathbf{Y}\, Q(y)) \equiv \forall y{\in}\mathbf{Y}\,(P(x) \vee Q(y))\\
&P(x) \rightarrow (\forall y{\in}\mathbf{Y}\, Q(y)) \equiv \forall y{\in}\mathbf{Y}\,(P(x) \rightarrow Q(y))\\
&P(x) \nleftarrow (\forall y{\in}\mathbf{Y}\, Q(y)) \equiv \forall y{\in}\mathbf{Y}\,(P(x) \nleftarrow Q(y)), \text{ provided that } \mathbf{Y} \neq \emptyset
\end{aligned}
$$

- For example, it is erroneous to state *"all persons are not married"* when it is meant that "not all persons are married“.

$$\neg\, \exists x{\in}\mathbf{X}\, P(x) \equiv \forall x{\in}\mathbf{X}\, \neg P(x) \not\equiv \neg\, \forall x{\in}\mathbf{X}\, P(x) \equiv \exists x{\in}\mathbf{X}\, \neg P(x)$$

where P(x) can be the predicate *“x is married”*

PREDICATE CALCULUS FUNCTIONS

- In contrast to the predicates, which maps the input objects to true or false, functions maps the input objects to an output object (constant).

$$y = function(a_1, a_2, \cdots, a_n).$$

- For example, the function

$$y = teacher('Peter').$$

assigns y to a teacher of Peter.

Predicate Inference Rules

- A rule of inference states that, given a particular formula (or set of formulas) with a certain property as a hypothesis, another specific formula (or set of formulas) can be derived as a conclusion.
- The rule is sound (or truth-preserving) if it preserves validity in the sense that whenever any interpretation satisfies the hypothesis, that interpretation also satisfies the conclusion (deductive reasoning).
- Predicate rules of inference can be derived directly from the inference rules of the propositional calculus, such as Conjunction Introduction, Modus Ponens, Resolution etc.

PREDICATE INFERENCE EXAMPLE

- The Modus Ponens rule can be state as

Premise 1: $P \rightarrow Q$

Premise 2: P

Consequent: Q

$$\frac{P \rightarrow Q, P}{\therefore Q}$$

- Example:

Premise 1: $(\forall x)(fish(x) \rightarrow swims(x))$

Premise 2: $fish(trout)$

Consequent: $swims(trout)$

$$\frac{(\forall x)(fish(x) \rightarrow swims(x))\ , fish(trout)}{\therefore swims(trout)}$$

Proving Inference Rules

- Inference formulas can be proved by showing that the negation of the formula is unsatisfiable (proof by contradiction or *Reductio ad absurdum*).
- A technique based on the resolution inference is commonly used in automated theorem proving.
- To recast the reasoning using the resolution technique, first the predicate clauses must be converted to conjunctive normal form (CNF) and a list of such clauses is formed from the premises.
- The negated original consequent (conclusion) is also added to the list of the CNF clauses.

Proving Inference Rules

- A formula is in conjunctive normal form (CNF) or clausal normal form if it is a conjunction of clauses, where a clause is a disjunction of literals (i.e., an AND of ORs).
- Pairs of such clauses are then selected, where the same predicate is negated in one clause and not negated in the other one.
- The modus ponens inference can then be applied as a special case of the resolution inference to infer a consequent clause from the two given premise clauses.

Premise 1: P

Premise 2: $\neg P \vee Q$

Consequent: Q

PROVING INFERENCE RULES

- In this way, the list of CNF clauses is processed and reduced unless it is empty or a contradiction is encountered.
- If a contradiction is encountered then it is assumed that the negated consequent of the inference rule was rejected and therefore the original (non-negated) consequent is proved to be valid/true.

PROVING INFERENCE RULE EXAMPLE

- Let us prove the following inference:

  Premise 1: $(\forall x)(fish(x) \rightarrow swims(x))$

  Premise 2: $fish(trout)$ **Consequent**: $swims(trout)$

- The negated consequent $\neg swims(trout)$ is added to the list of clauses, and the first premise is converted to the CNF clause.
- As $(\forall x)(fish(x) \rightarrow swims(x))$ is true for all x, it is true for $x = \text{trout}$ as well, i.e. the premise can be rewritten as $fish(trout) \rightarrow swims(trout)$.
- Using the equivalence $(P \rightarrow Q) \leftrightarrow (\neg P \vee Q)$, the first premise can be rewritten as $\neg fish(trout) \vee swims(trout)$ that is a valid CNF clause.

Proving Inference Rule Example

- The list of CNF clauses then contains the following clauses

$$\neg swims(trout),$$
$$\neg fish(trout) \vee swims(trout),$$
$$fish(trout).$$

- The second pair clauses are then selected, as the same predicate $fish(trout)$ is negated in one clause and not negated in the other one.

- The modified modus ponens inference (seen as a special case of the resolution inference) is then applied:

Premise 1: $fish(trout)$

Premise 2: $\neg fish(trout) \vee swims(trout)$

Consequent: $swims(trout)$

Proving Inference Rule Example

- The list of CNF clauses is the reduced to

$$\neg swims(trout),$$
$$swims(trout).$$

- The resolution inference rule is then applied on the last pair of clauses that is actually a contradiction:

Premise 1: $swims(trout)$

Premise 2: $\neg swims(trout)$

Consequent: $false$

- As the contradiction was encountered the original consequent $swims(trout)$ is proved to be $true$.

First-order logic

- The presented logical formal system is also known as the first-order logic.
- The adjective "first-order" distinguishes this logic from higher-order logic in which there are predicates having other predicates or functions as arguments, or in which one or both of predicate quantifiers or function quantifiers are permitted.
- The Prolog (PROgrammation en LOGique) is an example of a computer programming language that implement the first-order logical system.

QUESTIONS

- What are the types of logic?
- Give examples of meaningful logical propositions.
- What logical connectives are used with propositional logic?
- Which proposition is called a tautology and which a contradiction?
- What are the main rules of logical reasoning?
- What is the difference between predicates and propositions?
- What are the two fundamental quantifications in predicate logic?