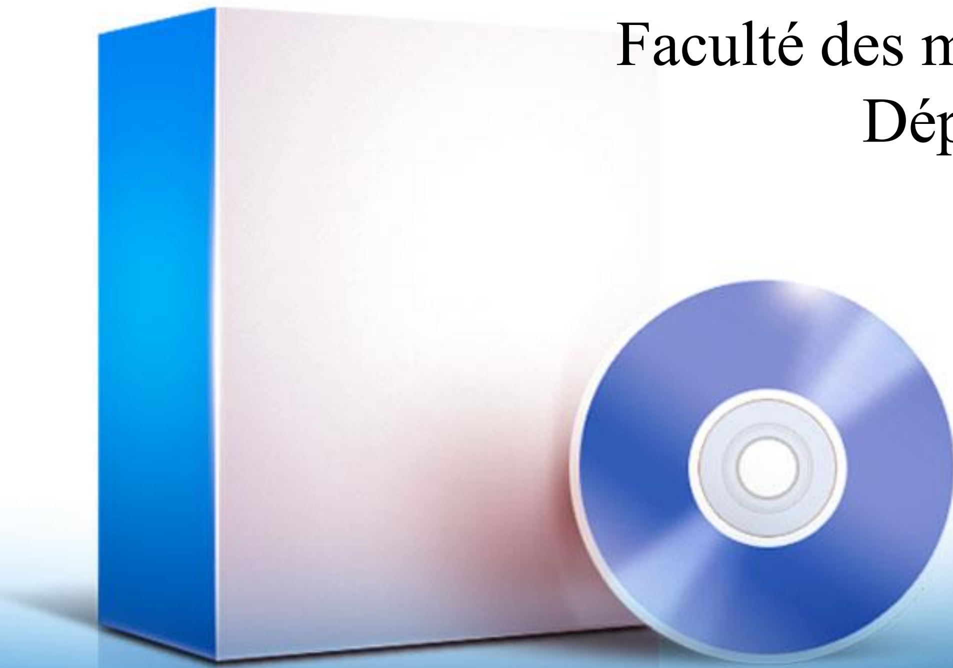
## 1 ER ANNÉE MASTER IA

Université de M'Sila
Faculté des mathématiques et de l'informatique
Département d'informatique
2025



1 ER ANNÉE MASTER IA

# 2025-2026

# 1-1-Propositional logic

# Introduction

**Proposition**

A proposition is any expression that can be true or false.

**Example**

- The sky is blue.
- The student is intelligent.
- Med is taller than Leila.

**propositions**

# Introduction

**Proposition**

A proposition is any expression that can be true or false.

**Example**

- Write your lesson.
- What is the distance between M'sila and Setif?
- Give me your birth date.

**Are not propositions**

# Propositional logic

Propositional logic is the mathematics of the two Boolean values true and false and the five operators whose names are:

# Propositional logic

Logical operators

$$\wedge \, Conjonction \; (and)$$

$$\vee \, Disjonction \; (or)$$

$$\neg \; Negation \; (Not)$$

$\rightarrow$ implication

$\equiv$ ($\leftrightarrow$)equivalence (Biconditional )

# Propositional logic

$x \wedge y \; is \; true \; iff \; x \; and \; y \; are \; both \; true$

$x \vee y \; is \; true \; iff x \; or \; y \; is \; true$

$\neg \, x \; is \; a \; true \; proposition \; iff \; x \; is \; false.$

$x \rightarrow y \; is \; true \; iff \; x \; is \; false \; or \; y \; is \; true.$

$x \equiv \; y \; is \; true \; iff \; are \; both \; true$
$or \; both \; false.$

# Propositional logic

Let's consider Op = { ¬, ∧, ∨, ⇒, ⇔, (, ) }

*Σ is a set of* ***symbols***. = {P, Q, R,…]

*The set of formulas of PL is defined recursively as follow:*

# Propositional logic

- *T* and *F* are two atomic formula (true et false).

  sometimes (T et ⊥).
- All variables (propositions) which are in $\Sigma$, are atomic formulas.
- if *P* and *Q* are formulas or atomic formulas then

$\neg P$, $(P)$, $P \wedge Q$, $P \vee Q$, $P \Rightarrow Q$, $P \Leftrightarrow Q$

are also formulas.

# The Truth table

A truth table is a mathematical table used in logic to calculate the truth value of a logical formula depending on its variables.

# The Truth table

Example : $P \rightarrow Q$

| **P** | **Q** | **$P \rightarrow Q$** |
|---|---|---|
| F | F | T |
| F | T | T |
| T | F | F |
| T | T | T |

# Example

Translate the following sentence into logical propositions :

If Samir doesn't work, then he won't get paid and his family will be sad.

# Exemple

We will start by identifying the words that are called logical words.

**If** Samir does**n't** work, **then** he **won't** get paid **and** his family will be sad.

# Example

**<u>If</u>** Samir does**n't** work, **<u>then</u>** he **won't** get paid **<u>and</u>** his family will be sad.

P : Samir works

Q : he get paid.

R : his family will be sad.

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

# interpretation

the mapping I : $\Sigma \rightarrow \{t, f\}$, which assigns to each propositional variable a truth value is called an interpretation.

So each formula has $2^n$ interpretations

# Eaemple

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

| P | Q | R | ¬P | ¬Q | ¬Q ∧ R | ¬P→(¬Q∧ R ) |
|---|---|---|---|---|---|---|
| F | F | F | T | T | F | F |
| F | F | T | T | T | T | T |
| F | T | F | T | F | F | F |
| F | T | T | T | F | F | F |
| T | F | F | F | T | F | T |
| T | F | T | F | T | T | T |
| T | T | F | F | F | F | T |
| T | T | T | F | F | F | T |

# A formula

A formula is called

- **Satisfiable**: if it is true for at least one interpretation.
- **Logically valid** or simply **valid**: if it is true for all interpretations; it is also called a **tautology**.
- **Unsatisfiable**: if it is not true for any interpretation.
- Each interpretation that satisfies a formula is called a **model** of that formula.

# Equivalences

implication

$$\neg A \lor B \Leftrightarrow A \Rightarrow B$$

contraposition

$$A \Rightarrow B \Leftrightarrow \neg B \Rightarrow \neg A$$

Equivalence

$$(A \Rightarrow B) \land (B \Rightarrow A) \Leftrightarrow (A \Leftrightarrow B)$$

# Equivalences

## Law of Morgan

| | | |
|---|---|---|
| $\neg(A \wedge B)$ | $\Leftrightarrow$ | $\neg A \vee \neg B$ |
| $\neg(A \vee B)$ | $\Leftrightarrow$ | $\neg A \wedge \neg B$ |

## Distributive

| | | |
|---|---|---|
| $A \wedge (B \vee C)$ | $\Leftrightarrow$ | $(A \wedge B) \vee (A \wedge C)$ |
| $A \vee (B \wedge C)$ | $\Leftrightarrow$ | $(A \vee B) \wedge (A \vee C)$ |

# Equivalences

Tautologie

$$A \lor \neg A \Leftrightarrow T$$

$$A \lor T \Leftrightarrow T$$

Contradiction

$$A \land \neg A \Leftrightarrow F$$

$$A \land F \Leftrightarrow F$$

# Equivalences

| | | |
|---|---|---|
| $A \lor F$ | $\Leftrightarrow$ | $A$ |
| $A \land T$ | $\Leftrightarrow$ | $A$ |

# Conjunctive Normal Form (CNF)

A formula is in **CNF** if it is expressed as a **conjunction (and) of disjunctions (AND)** of literals.

•**Literal**: an atomic proposition or the negation of an atomic proposition.

•**Clause**: a disjunction of literals.

•A formula is in **Conjunctive Normal Form (CNF)** if and only if it consists of a conjunction of clauses:

$$C_1 \land C_2 \land \cdots \land C_n.$$

# Conjunctive Normal Form (CNF)

A formula is in **DNF** if it is expressed as a **disjunction (OR) of conjunctions (AND)** of literals..

- term or minterm : a conjunction of literals.
- A formula is in *disjunctive normal form* DNF iff it consists of a **disjunction** of terms:

$$t_1 \vee t_2 \vee \cdots \vee t_n$$

# Conjunctive Normal Form (CNF)

We want to put in conjunctive normal form, the formula $(p \vee \neg (q \vee \neg(r \wedge \neg q))) \wedge (\neg p \vee q)$

Morgan's law : $(p \vee \neg (q \vee (\neg r \vee q))) \wedge (\neg p \vee q)$

Distribution of V : $(p \vee \neg (q \vee \neg r \vee q)) \wedge (\neg p \vee q)$

$x \vee x = x$ : $(p \vee \neg (q \vee \neg r)) \wedge (\neg p \vee q)$

# Example

$x \vee x = x$ : $(p \vee \neg (q \vee \neg r)) \wedge (\neg p \vee q)$

Law of Morgan : $(p \vee (\neg q \wedge r)) \wedge (\neg p \vee q)$

Distribution of $\vee$ : $((p \vee \neg q) \wedge (p \vee r)) \wedge (\neg p \vee q)$

finally : $(p \vee \neg q) \wedge (p \vee r) \wedge (\neg p \vee q)$

The formula is in Conjunctive Normal Form

# Conjunctive Normal Form (CNF)

**Theorem:** Any formula of propositional logic can be transformed into an equivalent formula in conjunctive normal form CNF.

# Inference Systems

## Logical Equivalence

Two formulas e and f are equivalent if and only if the formula $e \Leftrightarrow f$ is a tautology.

# Inference Systems

## Logical Equivalence

Are the two formulas $p \rightarrow q$ and $\neg q \rightarrow \neg p$ equivalent?

*To answer this question, we can use a truth table to verify that* $(p \rightarrow q) \leftrightarrow (\neg q \rightarrow \neg p)$.

## Logical Equivalence

$(p \rightarrow q) \leftrightarrow (\neg q \rightarrow \neg p)$.

| p | q | ¬q | ¬p | $p \rightarrow q$ | $\neg q \rightarrow \neg p$ | $(p \rightarrow q) \leftrightarrow (\neg q \rightarrow \neg p)$ |
|---|---|---|---|---|---|---|
| **F** | F | T | T | T | T | T |
| **T** | F | T | F | F | F | T |
| **F** | T | F | T | T | T | T |
| **T** | T | F | F | T | T | T |

# Inference Systems

## Logical Consequence

We say that A is a logical consequence of B if and only if every model of A is also a model of B.

In other words, in every model where A is true, B is also true.

$$A \vDash B \; iff \; M(A) \subseteq M(B)$$

# Inference Systems

## Using CNF for Satisfiability

### Davis-Putnam-Logemann-Loveland (DPLL) Algorithm

**DPLL Algorithm**
**Input:** A formula $F$ in CNF, a partial model $M$
**Output:** *false* if no model is found, *true* if a model is found + the model

1. If $F$ is empty:
   1. Return *true* and $M$
2. If $F$ contains an empty clause:
   1. Return *false*
3. For every unit clause $(x)$ or $(\neg x)$ in $F$:
   1. Set $M(x) = t$ (resp. *f*)
   2. $F$ = **Propagate Value**(x, $F$)
4. $K$ := choose a literal from $F$
5. Return **DPLL**($F \wedge K$) **or else DPLL**($F \wedge \neg K$)

# Inference Systems

## Using CNF for Satisfiability

Davis-Putnam-Logemann-Loveland (DPLL) Algorithm

**Value Propagation**

If we have set **M(x) = true**:

- Delete all clauses where **X** appears positively.
- Remove ¬X from all clauses where it appears.

**If we have set M(x) = false:**

- Delete all clauses where ¬X appears.
- Remove **X** from all clauses where it appears positively.

# Inference Systems

## Using CNF for Satisfiability

**Example :** $(p \vee q \vee \neg r) \wedge (\neg p \vee s) \wedge r \wedge (r \vee s)$

First, the formula is in Conjunctive Normal Form (CNF).

We will choose the literal r because it exists in the formula as a unit clause, and we set $I(r) = true$.

Therefore, we will delete all clauses that contain the literal r.

$(p \vee q \vee \neg r) \wedge (\neg p \vee s) \wedge r \wedge (r \vee s)$ becomes $(p \vee q \vee \neg r) \wedge (\neg p \vee s)$

Remove ¬r from all clauses where it appears.

$(p \vee q \vee \neg r) \wedge (\neg p \vee s)$ becomes $(p \vee q) \wedge (\neg p \vee s)$

# Inference Systems

## Using CNF for Satisfiability

At this step, we don't have a unit clause, so we choose between literals p and q we set *I(p)=t* and repeat the process : $(p \vee q) \wedge (\neg p \vee s)$ becomes $(s)$.

Finally, we are left with a single literal: $I(s) = t$, and we repeat

# Inference Systems

## Using CNF for Satisfiability

Delete s from the formula (s) therefore the formula becomes an empty formula.

Therefore the formula is satisfiable and its model for this formula $p=t$, $q=f$, $r=t$, $s=t$

# Inference Systems

## Using CNF for Satisfiability

At this step, we don't have a unit clause, so we choose between literals p and q we set *I(p)=t* and repeat the process : $(p \vee q) \wedge (\neg p \vee s)$ becomes $(s)$.

Finally, we are left with a single literal: $I(s) = t$, and we repeat

(s) therefore the formula becomes an empty formula.

Therefore the model for this formula $p=t$, $q= f$, $r = t$, $s=t$

# Inference Systems

## Resolution Method (Robinson's Method)

**The rule of *modus ponens*:** a simple and intuitive rule of inference which consists of starting from the validity of: A and A → B, and deducing B. It is denoted as:

$$\frac{\boldsymbol{A, A \rightarrow B}}{\boldsymbol{B}} \ \boldsymbol{ou}\ \boldsymbol{A, A \rightarrow B \vdash B}$$

# Inference Systems

**Resolution Method (Robinson's Method)**

The resolution rule is a generalization of the *modus ponens* rule, such that:

$$\frac{\boldsymbol{A} \vee \boldsymbol{B}, \neg \boldsymbol{B} \vee \boldsymbol{C}}{\boldsymbol{A} \vee \boldsymbol{C}}$$

$A \vee C$ Is called the **resolvent**

# Inference Systems

## Resolvent of two clauses

Let $C_1$ and $C_2$ be two clauses and $l$ be a literal, such that $l$ appears in $C_1$ and $\neg l$ appears in the other. The **resolvent** of $C_1$ and $C_2$ is the clause obtained by deleting $l$ and $\neg l$ from the two clauses and combining the remainder into a single clause.

Example : $C_1 = \neg B \vee C \vee D, C_2 = B \vee C \vee E$ therefore

$Res(C_1, C_2) = C \vee D \vee E$

# Inference Systems

## Resolvent of two clauses

**Question :** Prove that : $p \vee\ q \vee r, \neg p \vee q \vee r,\ \neg q \vee r\ \vDash r$

Let us demonstrate using the resolution method:

$$\{p \vee q \vee r, \neg p \vee q \vee r, \neg q \vee r\}$$

$$\{p, \neg p\}$$

$$\{q \vee r, \neg q \vee r\}$$

$$\{q, \neg q\}$$

$$\{r\}$$

# Inference Systems

## The substitution

f(p/g) denotes the formula in which p is replaced by g, such that this substitution is defined as follows:

- If the formula f contains only the propositional variable p, then f(p/g) = g.
- If the formula f contains only the propositional variable q (q≠p), then f(p/g) = q.
- If F is of the form ¬H, then the formula f(p/g) = ¬H(p/g).

# Inference Systems

## The substitution

If F is of the form $(f_1 \vee f_2)$, then the formula $f(p/g) = (f_1(p/g) \vee f_2(p/g))$

Si F is of the form $(f_1 \wedge f_2)$ then the formula $f(p/g) = (f_1(p/g) \wedge f_2(p/g))$

If F is of the form $(f_1 \rightarrow f_2)$, then the formula $f(p/g) = (f_1(p/g) \rightarrow f_2(p/g))$

If F is of the form $(f_1 \leftrightarrow f_2)$, then the formula $f(p/g) = (f_1(p/g) \rightarrow f_2(p/g))$

# Inference Systems

## Substitution

We say that a formula F is an instance of another formula G if F is obtained by substituting some propositional variables in G with formulas $F_i$.

$F = (A \to B) \to C$

$G = F(A/A \to C, C/C \land B)$

$G = ((A \to C) \to B) \to C \land B)$ We say that G is an instance of F.

# Inference Systems

## Logical Modeling

When modeling a system, the state of that system provides the interpretation of the variables.

Precisely, we represent certain properties of the system, which can be true or false, using logical variables (if there are *n* variables, we can represent $2^n$ states of the system).

# Inference Systems

## Logical Modeling

Example: We are interested in the safety system of an automobile
This system can be characterized by the following variables:

| variable | represented property |
|---|---|
| R | the engine is running |
| D1 | door 1 is open |
| D2 | door 2 is open |
| D3 | door 3 is open |
| D4 | door 4 is open |
| K | the ignition key is present |
| M | the car is moving |
| DA | the door alert is on |
| KA | the ignition key alert is on |

# Inference Systems

## Logical Modeling

**Question:** express in propositional logic

When the car is moving, all doors must be closed; otherwise, the door alert must be activated.

$$\{\mathrm{M} \wedge (\mathrm{D1} \vee \mathrm{D2} \vee \mathrm{D3} \vee \mathrm{D4})\} \Longrightarrow DA$$

# Inference Systems

## Logical Modeling

If the engine is off, door 1 is open, and the ignition key is present, then the key alert must be activated.

$$(\neg R \wedge D1 \wedge K) \Longrightarrow KA$$

The key alert should only be activated in this circumstance.

$$KA \Longrightarrow (\neg R \wedge \mathrm{D1} \wedge K)$$