University Kasdi Merbah Ouargla

3rd year engineering

Module: Fundamentals of AI

Lesson: Non Classical Logic

Prepared by : Mrs. AMRANE Leila

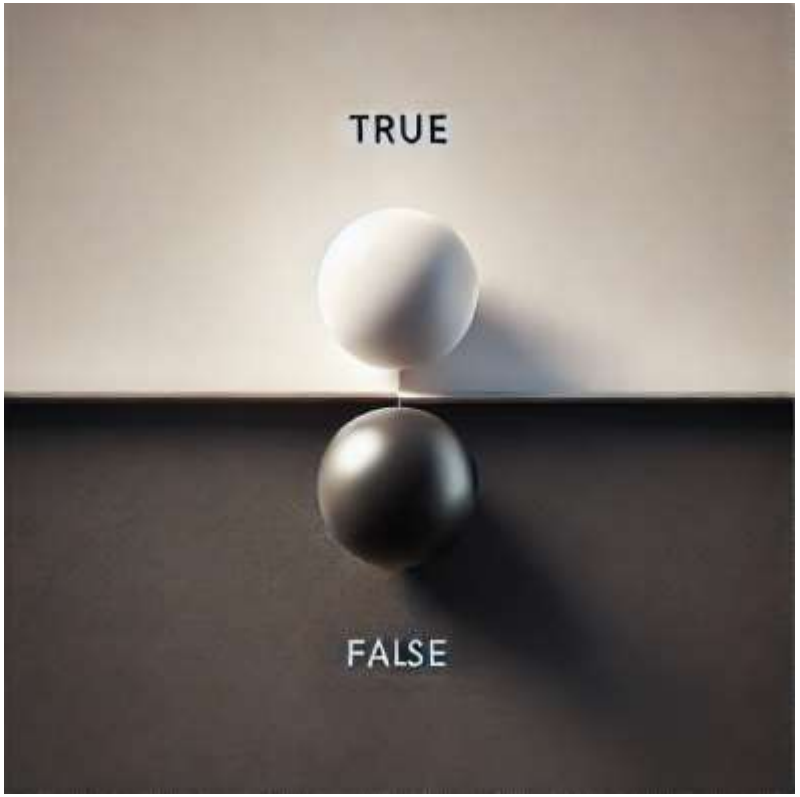
2025-2026

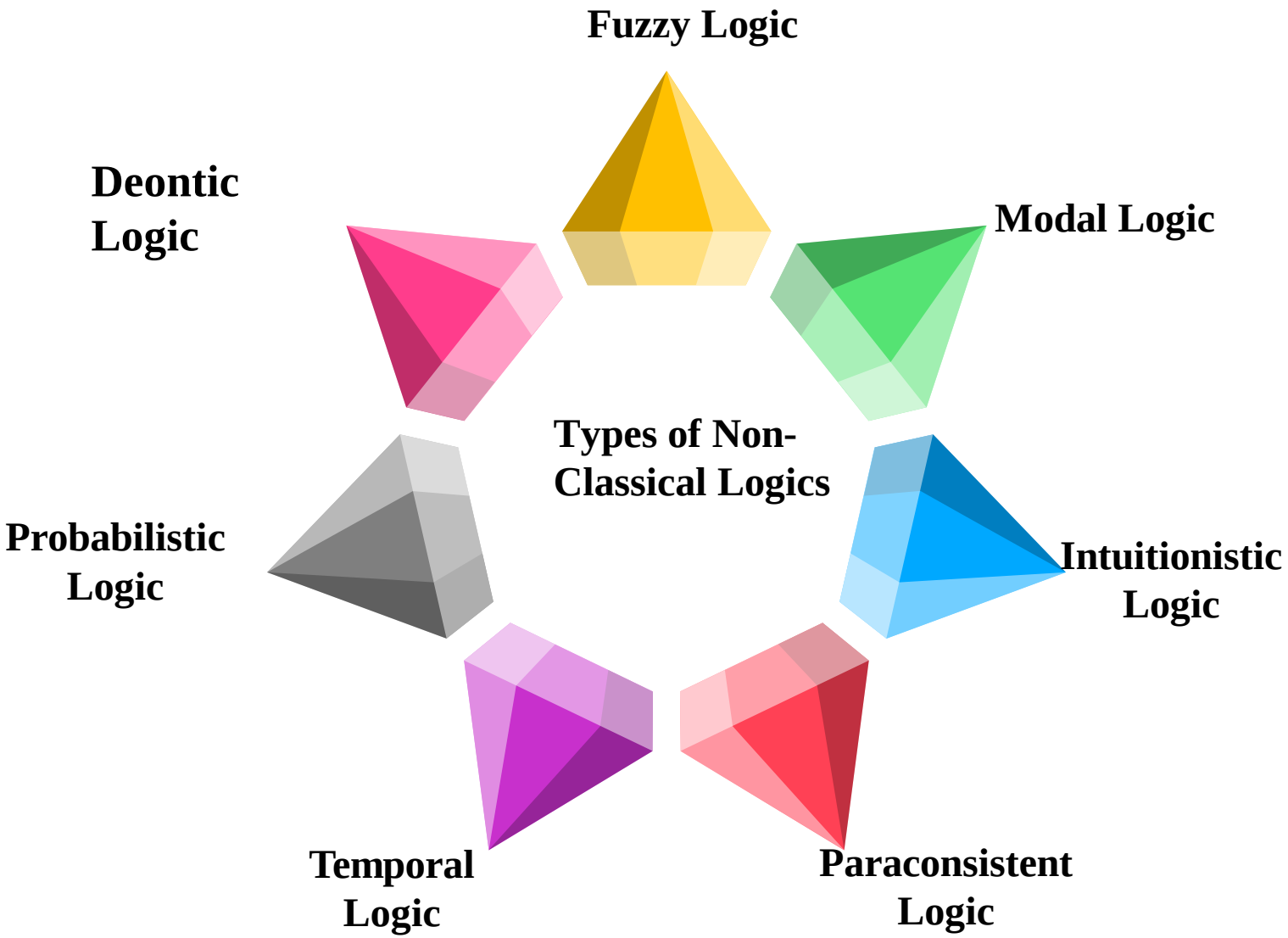
Non-Classical Logic

Classical logic is mainly based on two fundamental principles:

The **law of excluded middle**

The **law of non-contradiction**




Fuzzy Logic
Modal Logic
Intuitionistic Logic
Paraconsistent Logic
Temporal Logic
Probabilistic Logic
Deontic Logic
Types of Non-Classical Logics

Types of non-classical logics

Situation:

An automatic washing machine must adjust the amount of water according to the weight of the clothes.
Instead of making binary decisions (a lot of water or little water), the machine must choose a water level between **0% and 100%** depending on the quantity of clothes.

Which non-classical logic would you use to model this situation?

Fuzzy Logic

Fuzzy Logic

Principle:

Developed by **Lotfi Zadeh** in the 1960s, fuzzy logic allows reasoning with **degrees of truth** — representing **partial truth** rather than absolute true/false values.

Characteristics

- ❑ Models uncertainty and vague concepts.
- ❑ Each proposition has a degree of truth, allowing a smooth transition between true and false

Applications

- ❑ Control systems (washing machines, air conditioners, industrial automation).
- ❑ Artificial Intelligence systems handling uncertainty.

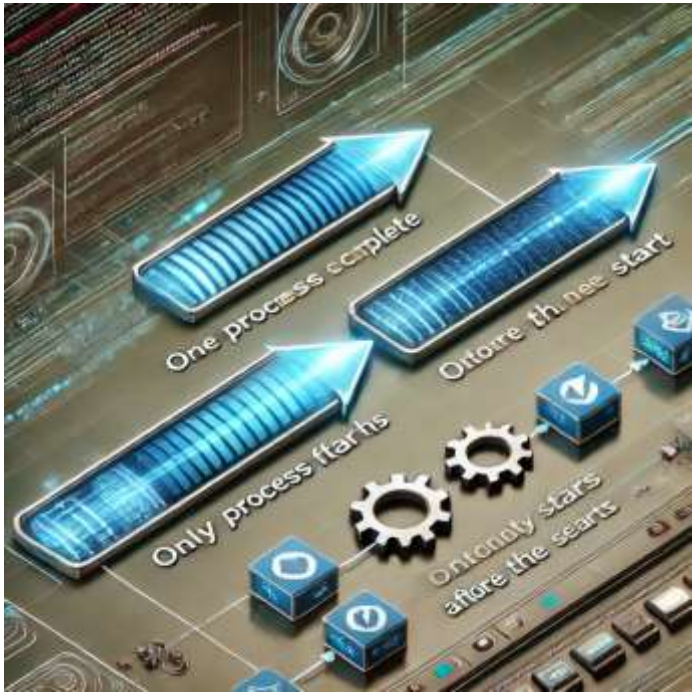
Types of non-classical logics

Situation:

Two computer processes must run in parallel, but one must only start **after** the other has finished.
You must ensure the scenario works correctly — the second process should **never** start before the first one ends.

Which non-classical logic would you use?

Temporal Logic



Temporal Logic

Principle:

Introduces operators such as:
F (Future), **G** (Globally / Always), **U** (Until).
It allows reasoning about **events over time**, such as “eventually” or “always”.

Characteristics

- Used to reason about **time-dependent systems** and verify behavior sequences.

Applications

- Formal verification of **software and hardware systems**.
- Used in **real-time** or **concurrent systems**.

Types of non-classical logics

Situation:

A meteorologist says there's a **70% chance of rain** tomorrow. You must plan an outdoor activity and decide whether to take precautions.

Which non-classical logic applies here?

Probabilistic Logic

Probabilistic Logic

Principle:

Combines **logic** with **probability theory** to handle uncertainty. Instead of assigning “true” or “false,” a **probability value** is given to each statement.

Characteristics

- ❑ Allows reasoning with uncertain or incomplete information.
- ❑ Logical operators incorporate probability degrees.

Applications

- ❑ Artificial Intelligence and Machine Learning.
- ❑ Decision-making systems in uncertain environments

Types of non-classical logics

Situation:

A database contains **contradictory information** about a person's birthdate — one record says 1980, another says 1985.
You don't want to reject the entire database because of this contradiction.

Which non-classical logic applies here?

Paraconsistent Logic

Paraconsistent Logic

Principle:

Accepts local contradictions **without collapsing** the entire logical system.

Characteristics

- ❑ Tolerates contradictions and prevents illogical conclusions.
- ❑ Enables reasoning within inconsistent data sources.

Applications

- ❑ Managing **inconsistent databases**.
- ❑ AI systems that must deal with contradictory information

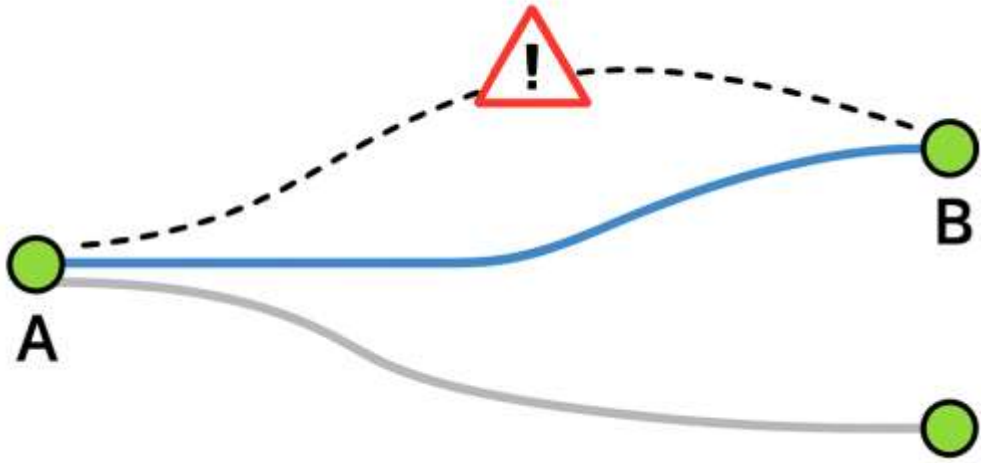
Types of non-classical logics

Situation:

You must travel from point A to point B, but several routes exist and one might be closed in case of bad weather.

You need to plan not only for the best route under normal conditions but also for possible closures.

Which non-classical logic helps model **possibility** and **necessity**?



Modal Logic

Modal Logic

Principle:

Modal logic extends classical logic by introducing **modal operators** that qualify statements as *necessary* or *possible*:
□**P** → "It is necessary that P."
◇**P** → "It is possible that P."
It allows reasoning about **different possible worlds** or **alternative scenarios**.
...

Applications

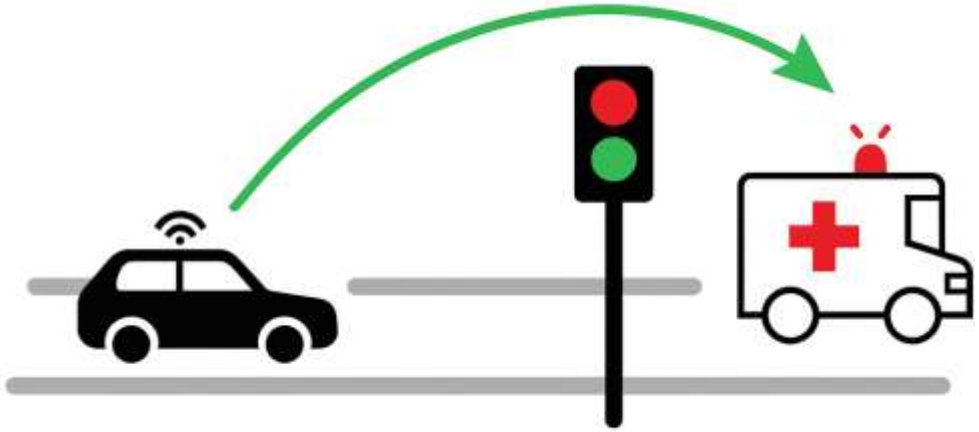
- **Philosophy and epistemology**: reasoning about necessity and knowledge.
- **Computer science**: program verification, temporal and dynamic systems.
- **Artificial Intelligence**: representing belief, knowledge, and possibilities in agents.

Types of non-classical logics

Situation:

A self-driving car must obey traffic laws but also react to emergencies, for example, allowing an ambulance to pass even if the light is red. This involves reasoning about **obligations**, **permissions**, and **prohibitions**.

Which non-classical logic helps express what is allowed or required?



Deontic Logic

Deontic Logic

Principle:

Deontic logic is a **branch of modal logic** that formalizes concepts of **duty, permission, and prohibition** using modal-like operators:
O(P) → "It is obligatory that P."
P(P) → "It is permitted that P."
F(P) → "It is forbidden that P."

Characteristics

- ❑ Focuses on **normative reasoning** — rules, rights, and ethical duties.
- ❑ Handles **exceptions** (e.g., emergency priority over normal rules).
- ❑ Can represent **conflicting duties** or **conditional permissions**.

Applications

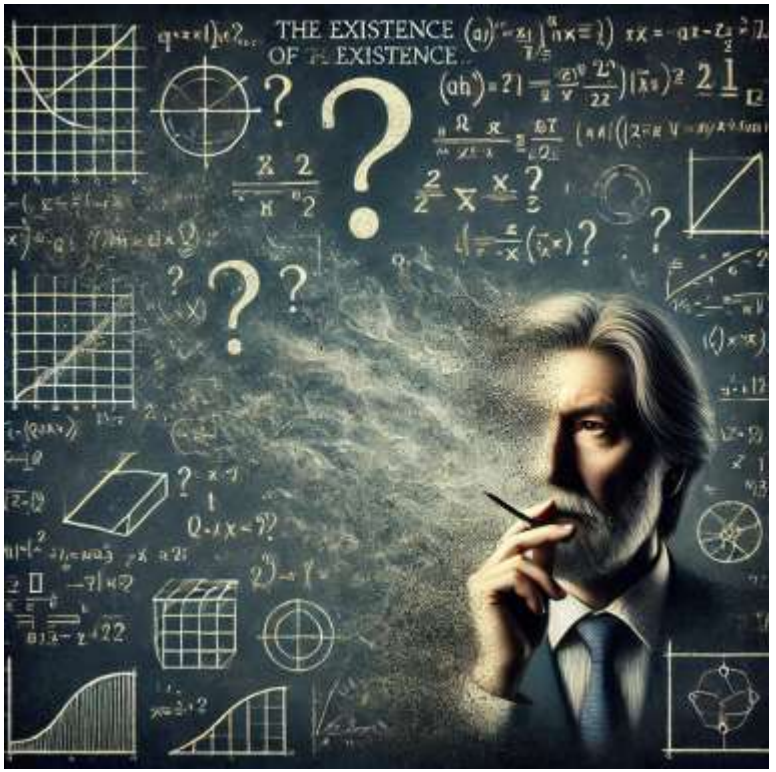
- ❑ **Law and ethics**: modeling legal or moral rules.
- ❑ **AI and autonomous systems**: managing priorities and ethical constraints.
- ❑ **Traffic and robotics**: defining behavior rules under exceptional conditions.

Types of non-classical logics

Situation:

In a constructive mathematical proof, you need to prove the existence of an object, but you don't yet have a direct constructive proof,

Which non-classical logic is most appropriate?



intuitionniste Logic

Intuitionniste Logic

Principle:

Rejects the **law of excluded middle**.
Without a constructive proof, you can't assert whether a statement is true or false.

Characteristics

- Requires **explicit proofs** for existence.
- A statement is true only if it can be **constructed**

Applications

- Constructive mathematics and **theoretical computer science**.
- Program verification systems and formal proofs.