AIKI100
Logic & AI

Department of Information Science and Media Studies
Universitetet i Bergen

September 24, 2025

What is logic?

What is logic?

- Studied since ancient times
- Connected with rhethorics and law

Just because something sounds *convincing*,
that doesn't mean it is *correct*.

...But what does it mean to be *correct*?

Hypotheses and inference

Arguments typically have the following structure:

$$\text{Hypotheses (Facts)} \xrightarrow{\text{Inference (Reasoning)}} \text{Conclusion}$$

The conclusion can be false if either a hypothesis is false or the inference is incorrect.
Logic is concerned with the study of *inference*.

A first example

An inference is typically written as:

$$\frac{\text{Premisses}}{\text{Conclusion}}$$

Meaning: whenever all the premisses are true, the conclusion must also be true.

A famous example:

$$\frac{\begin{array}{c}\text{All men are mortal}\\ \text{Socrates is a man}\end{array}}{\text{Socrates is mortal}}$$

Syllogisms: a definition

A syllogism is built from sentences of the form:

- "All Greeks are mortal"
- "Some Greek is mortal"
- "No Greek is mortal"
- "Some Greek is not mortal"

A syllogism is made of exactly two premisses and one conclusion, involving three terms ("Greeks", "mortals", "cats"...).

Examples of syllogisms

$$\frac{\text{All cats are mammals} \quad \text{All mammals are animals}}{\text{All cats are animals}}$$

$$\frac{\text{All cats are mammals} \quad \text{No bird is a mammal}}{\text{No bird is a cat}}$$

$$\frac{\text{All cats are mammals} \quad \text{Some animal is a cat}}{\text{Some animal is a mammal}}$$

$$\frac{\text{All cats are mammals} \quad \text{Some animal is not a mammal}}{\text{Some animal is not a cat}}$$

$$\frac{\text{No bird is a cat} \quad \text{Some animal is a bird}}{\text{Some animal is not a cat}}$$

$$\frac{\text{No bird is a cat} \quad \text{Some animal is a cat}}{\text{Some animal is not a bird}}$$

More examples of syllogisms (1)

$$\frac{\text{All rare things are expensive} \quad \text{A cheap horse is rare}}{\text{A cheap horse is expensive}}$$

The *inference* is correct, but one of the *premisses* is not.

NOT syllogisms

$$\frac{\text{All cats are mammals} \quad \text{All mammals are mortal}}{\text{All mortals are cats}}$$

$$\frac{\text{All cats are mammals} \quad \text{All dogs are mammals}}{\text{All dogs are cats}}$$

The *premisses* are true, but the *inferences* are incorrect.

More examples of syllogisms (2)

$$\frac{\text{All Greeks are men} \quad \text{All men are mortal}}{\text{All Greeks are mortal}}$$

The conclusion is true and the inference is correct. What about the premisses?

- The truth of a statement depends on the meaning of the words (which can be ambiguous).
- The correctness of an inference doesn't!

Abstracting terms

What matters is not the terms used, but the *structure* of the sentences:

$$\frac{\text{All P are Q} \quad \text{All Q are R}}{\text{All P are R}}$$

$$\frac{\text{All P are Q} \quad \text{No R is Q}}{\text{No R is P}}$$

$$\frac{\text{All P are Q} \quad \text{Some R is P}}{\text{Some R is Q}}$$

$$\frac{\text{All P are Q} \quad \text{Some R is not Q}}{\text{Some R is not P}}$$

$$\frac{\text{No P is Q} \quad \text{Some R is a P}}{\text{Some R is not Q}}$$

$$\frac{\text{No P is a Q} \quad \text{Some R is Q}}{\text{Some R is not P}}$$

Beyond basic syllogisms

What about more complex premisses?

$$\frac{\begin{array}{c}\text{All integers are either even or odd}\\ \text{If } x \text{ is an even integer, then } x+x \text{ is even}\\ \text{If } x \text{ is an odd integer, then } x+x \text{ is even}\end{array}}{\text{If } x \text{ is an integer, then } x+x \text{ is even}}$$

i.e.

$$\frac{\begin{array}{c}\text{All } P \text{ are } Q \text{ or } R\\ \text{All } Q \text{ are } S\\ \text{All } R \text{ are } S\end{array}}{\text{All } P \text{ are } S}$$

Some standard logics

Connectives and propositional logic

Ingredients of the propositional language:

- A set of **propositional variables** p, q, ...
- Some **logical connectives**:
  - $\neg$: NOT
  - $\wedge$: AND
  - $\vee$: OR
  - $\rightarrow$: IMPLIES
  - Also: $\leftrightarrow$ (equivalent to), XOR (exclusive "or")...

Formulas are typically called φ, ψ...

Truth tables

The relationship between the truth of a propositional formula and the truth of the variables appearing in it is formally defined. One way to write this definition is with **truth tables**:

φ	$\neg\varphi$
0	1
1	0

Meaning:

- If φ is false (0), then $\neg\varphi$ is true (1)
- If φ is true (1), then $\neg\varphi$ is false (0)

Truth tables for the other connectives

φ	ψ	$\varphi \land \psi$
0	0	0
0	1	0
1	0	0
1	1	1

φ	ψ	$\varphi \lor \psi$
0	0	0
0	1	1
1	0	1
1	1	1

φ	ψ	$\varphi \to \psi$
0	0	1
0	1	1
1	0	0
1	1	1

φ	ψ	$\varphi \leftrightarrow \psi$
0	0	1
0	1	0
1	0	0
1	1	1

φ	ψ	$\varphi \,\mathsf{XOR}\, \psi$
0	0	0
0	1	1
1	0	1
1	1	0

Using truth tables

Truth tables can be used to examine complex formulas:

p	q	$\neg p$	$\neg q$	$\neg p \vee \neg q$	$\neg(\neg p \vee \neg q)$
0	0	1	1	1	0
0	1	1	0	1	0
1	0	0	1	1	0
1	1	0	0	0	1

p	q	$p \wedge q$	$\neg(\neg p \vee \neg q)$	$(p \wedge q) \leftrightarrow \neg(\neg p \vee \neg q)$
0	0	0	0	1
0	1	0	0	1
1	0	0	0	1
1	1	1	1	1

Satisfiability and validity

p	q	$p \wedge q$	$\neg(\neg p \vee \neg q)$	$(p \wedge q) \leftrightarrow \neg(\neg p \wedge \neg q)$
0	0	0	0	1
0	1	0	0	1
1	0	0	0	1
1	1	1	1	1

A formula is:

- *Valid* if it is always true
  - Only 1's in its column of a truth table
  - Example: $(p \wedge q) \leftrightarrow \neg(\neg p \vee \neg q)$ is valid.
- *Satisfiable* if it is not always false
  - At least one 1 in its column of a truth table
  - Example: $p \wedge q$ is satisfiable.
- *Unsatisfiable* if it is always false (not satisfiable)
  - Only 0's in its column of a truth table

One more example

p	q	r	$p \wedge q \wedge r$	$p \wedge q \wedge \neg r$	$(p \wedge q \wedge r) \vee$ $(p \wedge q \wedge \neg r)$	$p \wedge q$
0	0	0	0	0	0	0
0	0	1	0	0	0	0
0	1	0	0	0	0	0
0	1	1	0	0	0	0
1	0	0	0	0	0	0
1	0	1	0	0	0	0
1	1	0	0	1	1	1
1	1	1	1	0	1	1

Two formulas are *logically equivalent* if they have the same truth table. This can be used to simplify formulas.

A previous example, the propositional way

$$\frac{\begin{array}{c}\text{All } P \text{ are } Q \text{ or } R\\ \text{All } Q \text{ are } S\\ \text{All } R \text{ are } S\end{array}}{\text{All } P \text{ are } S}$$

Translated to the propositional language:

$$\frac{p \to (q \vee r) \qquad q \to s \qquad r \to s}{p \to s}$$

Connectives in programming

If you program, you might know the connectives with different symbols:

```
while(monsterLife > 0 && playerLife > 0){...
keepgoing = !DeckEmpty && errors < 3;
if((WASD && input == 'W') || (Numpad && input == 5){...
```

"Formulas" in logic are the same thing as "booleans" in programming.

Here DeckEmpty, WASD, Numpad are the *propositional variables*: they are stored as their truth value (true or false) and have no further structure.

What about the other blue elements?

Adding structure: predicates

"`monsterLife` $>$ 0" or "x is an even integer" describe a property of an object that is *not* a formula (`monsterLife`, x).

"`input` $==$ '`W`'" and "`errors` $<$ 3" describe relationships between objects that are also not formulas (`input` and '`W`', `errors` and 3).

To go beyond propositional logic, we enrich the language with:

- A set of **variables** x, y, z...
  - But also `monsterLife`, `input`, '`W`', `errors`, 3,...
- **Predicate** symbols P, Q, R...
  - But also StrictlyPositive, Integer, Even, Odd, $=$, LesserThan...

Predicates: some examples

"If x is an integer, then x is either even or odd":

$$\mathrm{Integer}(x) \rightarrow \mathrm{Even}(x) \vee \mathrm{Odd}(x)$$

"The deck is not empty and we haven't received 3 error tokens":

$$\neg \mathrm{DeckEmpty} \wedge \mathrm{StrictlyLesserThan}(\mathit{errors}, 3)$$

"If you are my family and she is your family then she is my family":

$$\mathrm{Family}(\mathit{me}, \mathit{you}) \wedge \mathrm{Family}(\mathit{you}, \mathit{her}) \rightarrow \mathrm{Family}(\mathit{me}, \mathit{her})$$

Quantification

Compare:

- "If x is an integer, then x is either even or odd"
- "All integers are either even or odd"

Do both these sentences say the same thing?

- The first is talking about a particular x.
- The second says that the first sentence is true no matter what x is.

Let us add to our language some **quantifiers**:

- $\forall$: "for all" (universal quantifier)
- $\exists$: "there exists" (existential quantifier)

Quantification: some examples

"If x is an integer, then x is either even or odd":

$$\text{Integer}(x) \rightarrow \text{Even}(x) \vee \text{Odd}(x)$$

"All integers are either even or odd":

$$\forall x.(\text{Integer}(x) \rightarrow \text{Even}(x) \vee \text{Odd}(x))$$

"I have a favorite subject"

$$\exists x.(\text{Subject}(x) \wedge \forall y.(\text{Subject}(y) \rightarrow \text{Prefer}(x, y)))$$

"I have one cat and one only"

$$\exists x.(\text{MyCat}(x) \wedge \forall y.(\text{MyCat}(y) \rightarrow (x = y)))$$

Math formulas are logic formulas!

We could write everything as $P(x, y)$, $Q(x, y)$... but sometimes different notations are easier to read:

- $x = y$ rather than $=(x, y)$ or $\mathrm{Equals}(x, y)$
- $x > y$ rather than $>(x, y)$ or $\mathrm{GreaterThan}(x, y)$
- etc.

All math formulas are just logic formulas with those standard notations!

$$\forall x.\forall y.\forall z.(((x < y) \wedge (y < z)) \rightarrow (x < z))$$

Terms and functions

What about "the product of two negatives is a positive"?

$$\forall x . \forall y . ((x < 0) \wedge (y < 0)) \rightarrow (x * y > 0))$$

What is $x * y$ here?

- Built from two variables x and y
- Not a variable
- Not a formula (not true or false)...

Just like we gave propositional variables some more structure with predicates (from p to $P(x, y)$), we can give variables some more structure with **functions**: from x to $f(x, y)$ (or $x * y$...).

Terms are objects built from variables and function symbols. Predicates take terms as arguments.

Adding terms: some examples

"I am younger than my father":

$$age(me) < age(father(me))$$

or also

$$\mathrm{YoungerThan}(me, father(me))$$

This is a typical database query. Which formalization to go for also depends on what kind of data we have.

"If x is an integer then $x + x$ is even":

$$\mathrm{Integer}(x) \rightarrow \mathrm{Even}(x + x)$$

Back to that example

$$\frac{\begin{array}{c}\text{All integers are either even or odd}\\ \text{If } x \text{ is an even integer, then } x+x \text{ is even}\\ \text{If } x \text{ is an odd integer, then } x+x \text{ is even}\end{array}}{\text{If } x \text{ is an integer, then } x+x \text{ is even}}$$

In first-order logic:

$$\frac{\begin{array}{c}\forall x.(\mathrm{Integer}(x) \rightarrow (\mathrm{Even}(x) \vee \mathrm{Odd}(x)))\\ \forall x.((\mathrm{Integer}(x) \wedge \mathrm{Even}(x)) \rightarrow \mathrm{Even}(x+x))\\ \forall x.((\mathrm{Integer}(x) \wedge \mathrm{Odd}(x)) \rightarrow \mathrm{Even}(x+x))\end{array}}{\forall x.(\mathrm{Integer}(x) \rightarrow \mathrm{Even}(x+x))}$$

Logic and semantics

What is *a* logic? (in the mathematical sense)

- A language
- A set of **axioms**
- A set of **inference rules**

Some examples:

- $\varphi \to \varphi$ is an *axiom*.
- $\dfrac{\varphi \to \psi \qquad \varphi}{\psi}$ is a *rule*.
- $\dfrac{P(a)}{\exists x.P(x)}$ is also a rule.

Theorems

A **theorem** of a logic is a formula of that language that can be proved with only those axioms and rules (no additional hypotheses):

- All axioms are theorems
- If the premisses of an inference rule are all theorems, then its conclusion is also a theorem.

Note the difference between the axiom $\varphi \to \psi$ (whenever φ is true, so is ψ) and the rule $\frac{\varphi}{\psi}$ (if φ is ALWAYS true then so is ψ).

The meaning of the symbols **does not matter**: this is a mathematical system.

Theorems and validities

Examples:

- $(p \wedge q) \leftrightarrow \neg(\neg p \vee \neg q)$ is a theorem of propositional logic.
- $\forall x.(\mathrm{Integer}(x) \rightarrow (\mathrm{Even}(x) \vee \mathrm{Odd}(x)))$ is **not** a theorem of first-order logic: it is only valid for a specific **interpretation** of the predicates Integer, Even and Odd.

Often, a logic is accompanied by a specific **semantics**: an interpretation of the language such that any formula is a **theorem of that logic** if and only if it is **valid for that semantics**.

Truth tables are an example of semantics for predicate logic: they explicitly deal with the truth values of formulas.

Logic in Computer Science and AI

Logic in Computer Science (1)

- Everything is logic
- Programming uses logic (as we've already seen)
- Typing is logic: compare

$$\frac{\varphi \to \psi \qquad \varphi}{\psi} \quad \text{and} \quad \frac{f : \mathtt{INT} \to \mathtt{BOOL} \qquad x : \mathtt{INT}}{f(x) : \mathtt{BOOL}}$$

- Complexity analysis:
  - Problems have an inherent complexity and can be sorted into complexity classes
  - A very famous complexity class: NP (nondeterministic polynomial: a solution can be checked in polynomial time)
  - A very famous NP problem: SAT (satisfiability of propositional formulas of a certain form)

Logic in Computer Science (2)

- Processors:
  - Create outputs from a bunch of electrical inputs
  - An electrical input is: on or off
  - Or: 0 or 1
  - Or: true or false
  - How do we tell the processor which combination of 0s and 1s are supposed to give which output (also a 0 or 1)?
  - With logic!
  - Technically: take the formula corresponding to the input-output relation you want
  - Connectives become logic gates (a circuit turning inputs into the output corresponding to that connective)
  - Formula becomes big circuit full of logic gates
  - Logic becomes computer
- Everything is logic

Logic and AI

Some uses of logic in AI:

- (Some!!) theorem proving can be automated
- Proof checkers/assistants can be implemented (Coq, Isabelle...)
- Describing and querying data: ontologies, description logics
- Logic programming: Prolog
  - (Basically) Define a database (in particular, properties of objects, relationships between objects, functions on objects)
  - Check formulas

Logic in AI: planning

A planning task is:

- **Input:**
  - An initial state
  - A set of actions
  - A goal formula/set of goal states (same thing)
- **Output:** Is there a sequence of actions that will lead from the initial state to a goal state?

Planning is used in:

- Warehouse robots
- Search & rescue robots
- Video game NPCs
- etc.

Planning is logic (because everything is logic)

Everything in planning (states, action conditions and effects, goals) is described with logic!

> Everytime you need to
> formally describe some properties,
> you do that with logic

Also, any representation of data is a semantics for the formal language used to describe it. Studying the inherent properties of that representation is logic

Planning is logic (because it's model-checking)

Planning algorithms usually rely on some sort of graph search, where the graph represents possible states and state transitions.

But: if the logic is *dynamic* (i.e. you can talk about the results of actions within the logic), the plan existence problem becomes a model checking problem:

- Write a formula describing: after a non-deterministic choice of a sequence of actions, the goal is satisfied
- Is this formula true in the initial state?
- If so, then there is a plan

Complex agents and modal logics

Recently, AI has been interested in more complex agents, that can have:

- Knowledge
- Beliefs
- Preferences
- Obligations (e.g. ethical rules)
- Etc.

These things can be described with a special family of logics called **modal logics**.

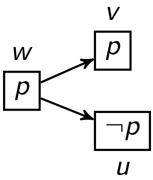
Modal logic: the basics

In modal logics, there are two **modal operators** $\Box$ and $\Diamond$.

- $\Box\varphi$: "necessarily, φ"
- $\Diamond\varphi$: "possibly, φ"

The (standard) associated semantics is called **Kripke semantics**.
It is a graph semantics:

- $\Box\varphi$ is true in w (written $w \models \Box p$) if φ is true in *all* accessible worlds from w
- $\Diamond\varphi$ is true in w if φ is true in *at least one* accessible world from w



Here, $w \not\models \Box p$ and $w \models \Diamond p$.

Interpretations of modal logics

These operators and semantics can be interpreted in many different ways:

- Temporal: $\mathrm{F}\varphi =$ "at every point in the future, φ"
- Spatial: $\mathrm{R}\varphi =$ "in any place on my right, φ" (add more operators for more dimensions, and axioms to describe their interactions!)
- Epistemic: $\mathrm{K}\varphi =$ "φ is known" $=$ "in any possible world, φ"
- Doxastic: $\mathrm{B}\varphi =$ "φ is believed" $=$ "in any (most) plausible world, φ"
- Deontic: $\mathrm{O}\varphi =$ "φ is obligatory" $=$ "in any allowed world, φ"
- etc.

More on epistemic logic

Let's take the example of epistemic logic:

$$\begin{aligned} \mathsf{K}\varphi &= \text{“}\varphi \text{ is known”} \\ &= \text{“}\varphi \text{ is true in any possible world”} \end{aligned}$$

Multi-agent version: $\mathsf{K}_a\varphi =$ “agent *a* knows that φ”

Basic axiomatics:

$$\left.\begin{array}{l} \blacksquare\ \mathsf{K}(\varphi \rightarrow \psi) \rightarrow (\mathsf{K}\varphi \rightarrow \mathsf{K}\psi) \\ \blacksquare\ \dfrac{\varphi}{\mathsf{K}\varphi} \end{array}\right\} \text{standard for any modal logic}$$

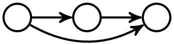
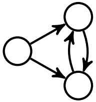
- $K\varphi \rightarrow \varphi$ (truth axiom)

Axioms and frame properties

Other possible properties:

- $K\varphi \rightarrow KK\varphi$: if you know something, then you know that you know it
- $\neg K\varphi \rightarrow K\neg K\varphi$: if you *don't* know something, then you know that you don't know it

Semantically, these correspond to properties on the relations:

$K\varphi \rightarrow \varphi$	Reflexivity	
$K\varphi \rightarrow KK\varphi$	Transitivity	
$\neg K\varphi \rightarrow K\neg K\varphi$	Euclideanness	

An application of epistemic logic: Hanabi

Hanabi is a cooperative card game where:

- Players hold their cards facing away from them
- The goal is to cooperatively play cards in a certain order
- Players communicate through limited, codified hints

This requires higher-order epistemic reasoning:

- What does this other player not know, and need to know?
- What does this other player's actions say about what they know?

Standard tree search algorithms like the ones used to play chess or Go aren't enough: we need epistemic logic!

Conclusion

Windup: what we learned today

- The aim of logic is to formally define correctness
- For that, we need a formally defined language
- From that language, we can specify acceptable rules of reasoning
- The rules reflect the correspondence between the language and the concepts and objects it seeks to describe
- Logic is used in many different fields of computer science
- In AI, logic gives us a framework in which to define agents and ensure that they behave in an expected way

A conclusion: on baroque and beautiful things

Baroque is an art style from 17th century Europe.

A conclusion: on baroque and beautiful things

The term "baroque" (probably) comes from "baroco", which is... a type of syllogism.

In 17th century France, "baroco" had become slang for "unnecessarily complicated", and was also used to refer to the newly emerging art style.



A conclusion: on baroque and beautiful things

Just like baroque art, whether you find logic beautiful or unnecessarily complicated is up to you!