Formal Logics

Introduction

Adrien Pommellet

March 27, 2025

What is this class about?

Logics. A coherent mode of reasoning that allows one to assess the truth of statements.

A Brief History of Logics

Blame the Greeks

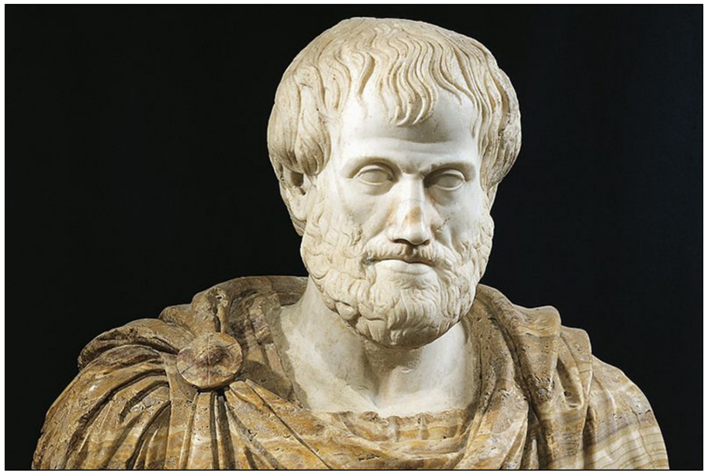


Figure 1: Aristotle (384–322 BC).

A Brief History of Logics

Aristotle's syllogism

- All men are mortal.
- Socrates is a man.
- Thus Socrates is mortal.

Figure 2: Gottfried Wilhelm Leibniz (1646–1716).

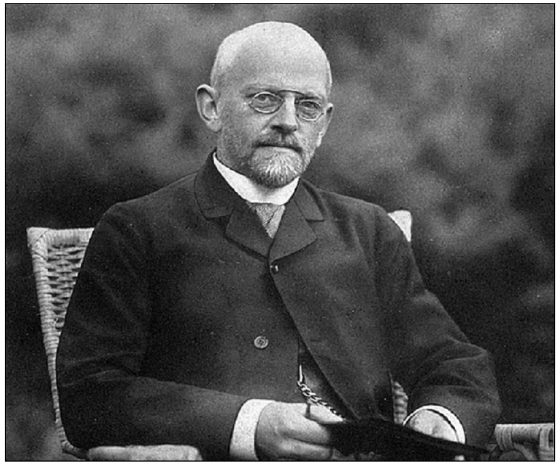


Figure 3: David Hilbert (1862–1943).

Syntax. The structural analysis of statements expressed as formulas. Truth is what we **build** using proofs.

Semantics. Interpreting formulas according to a mathematical model. Truth is a pre-existing absolute meant to be **discovered**.

Are these two notions equivalent?

Course Overview
A brief primer

- The full course consists in six two hour long **lectures**.
- The slides and detailed class notes can be found on **Moodle**. Beware of updates!
- You may ask questions in-between classes on a dedicated **Moodle** forum.

Course Overview
The dreaded exams

- A short mid-term exam (**7 points**) + a final exam (**13 points**).
- Make sure that you can properly access **Moodle Exam**.
- A short mock exam will allow you to get used to the interface.

Good luck!

Formal Logics
Propositional Logic

Adrien Pommellet

March 27, 2025

Of Induction
Inductive definitions I

Constructing arbitrarily complex objects from simpler ones through the means of fixed rules.

Of Induction
Inductive definitions II

Inductive definition of a set $\mathcal{T}$

It features three things:

Atomic objects. A set $\mathcal{A}$.

Constructors. A set $\mathcal{C}$; to each $op \in \mathcal{C}$, we match its **arity** $ar(op) \in \mathbb{N}$.

Depth. $d \in \mathbb{N} \cup \{\infty\}$.

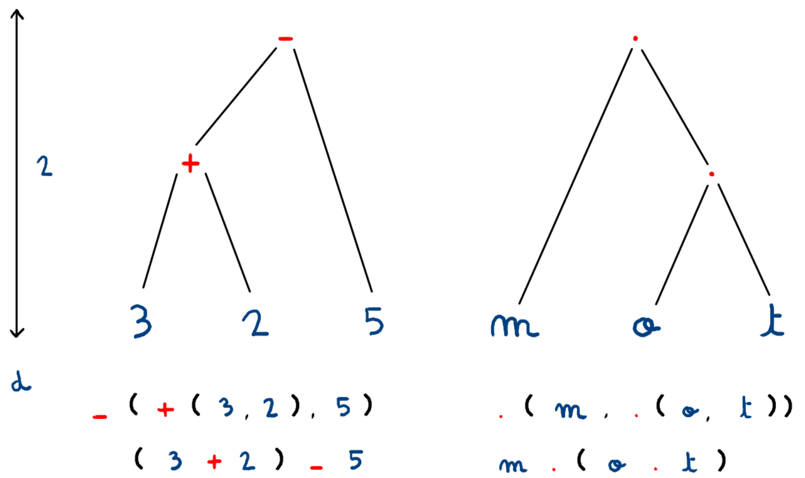
Starting from the atoms $\mathcal{A}$, we add new elements to $\mathcal{T}$ by combining existing elements using constructors in $\mathcal{C}$, allowing a **nesting depth** of at most d (or finite but unbounded if $d = \infty$).

Of Induction
Two examples I

Arithmetic terms	Words Σ^*
Atoms. $\mathbb{N}$.	Atoms. $\Sigma \cup \{\varepsilon\}$.
Rules. $\{+^2, -^2\}$.	Rules. $\{\cdot^2\}$.
Depth. ∞.	Depth. ∞.

The exponent ^{2} stands for the **arity** of each rule. Depth ∞ means that rules can be applied an unbounded but still **finite** number of times.

Of Induction
Two examples II



Of Induction

An inductive definition of depth

We can define **functions** on $\mathcal{T}$ such as depth inductively as well:

Depth of a term

Given an inductively defined set $\mathcal{T}$, we will define the **depth** function $\delta_{\mathcal{T}} : \mathcal{T} \to \mathbb{N}$ as follows:

On $\mathcal{A}$. For each atom $a \in \mathcal{A}, \delta_{\mathcal{T}}(a) = 0$.

Using $\mathcal{C}$. For each $op \in \mathcal{C}$ such that $ar(op) = n$ and $(t_1, \ldots, t_n) \in \mathcal{T}^n$, $\delta_{\mathcal{T}}(op(t_1, \ldots, t_n)) = \max(\{\delta_{\mathcal{T}}(t_1), \ldots, \delta_{\mathcal{T}}(t_n)\}) + 1$.

Intuitively, $\delta_{\mathcal{T}}(t)$ is the depth of t's **syntactic tree**.

Of Induction

Proof by structural induction

Goal. Given an inductively defined set $\mathcal{T}$ and a predicate $\mathcal{P}$, prove that $\forall x \in \mathcal{T}, P(x)$ holds.

We are going to use a proof by **structural** induction.

Base case. Prove that $\forall a \in \mathcal{A}, \mathcal{P}(a)$ holds.

Inductive case. For each constructor $op \in \mathcal{C}$, prove that if $ar(op) = n$ and $t_1, \ldots, t_n \in \mathcal{T}$ are n terms such that $\mathcal{P}(t_i)$ holds for any $i \in \{1, \ldots n\}$, then $\mathcal{P}(op(t_1, \ldots, t_n))$ holds.

Practical Application

Exercise 1. Prove that an arithmetic expression (as defined inductively earlier) with n operators always features $n + 1$ integers.

Answer

Propositional Formulas
An inductive definition

Propositional formulas

The set $\mathcal{F}_0 = \mathcal{F}_{\{\top,\bot,\neg,\wedge,\vee,\Rightarrow,\Leftrightarrow\}}$ is defined inductively as follows:

- $\mathcal{A}$. $\mathcal{V} \cup \{\top, \bot\}$ where $\mathcal{V}$ is a set of **variables**.
- $\mathcal{C}$. $\{\neg^1, \wedge^2, \vee^2, \Rightarrow^2, \Leftrightarrow^2\}$.
- d. ∞.

E Consider $(A \wedge (\neg B)) \Rightarrow C \in \mathcal{F}_0$.

Propositional Formulas
Valuations

Valuation

It is a function $\nu : \mathcal{V} \to \{\texttt{true}, \texttt{false}\}$.

Truth assignment function

Given a valuation ν, it is a function $|\ |_\nu : \mathcal{F}_0 \to \{\texttt{true}, \texttt{false}\}$.

Propositional Formulas
Semantics

Tarski's semantics

Defined inductively as follows:

- $|\top|_\nu = \texttt{true}$.
- $|\bot|_\nu = \texttt{false}$.
- Given $x \in \mathcal{V}$, $|x|_\nu = \nu(x)$.
- Given $\varphi \in \mathcal{F}_0$, $|\neg\varphi|_\nu = \texttt{true}$ if and only if $|\varphi|_\nu = \texttt{false}$.
- Given $\varphi, \psi \in \mathcal{F}_0$.
  - $|\varphi \vee \psi|_\nu = \texttt{true}$ if and only if $|\varphi|_\nu = \texttt{true}$ or $|\psi|_\nu = \texttt{true}$.
  - $|\varphi \wedge \psi|_\nu = \texttt{true}$ if and only if $|\varphi|_\nu = \texttt{true}$ and $|\psi|_\nu = \texttt{true}$.
  - $|\varphi \Rightarrow \psi|_\nu = \texttt{true}$ if and only if $|\varphi|_\nu = \texttt{true}$ implies $|\psi|_\nu = \texttt{true}$.
  - $|\varphi \Leftrightarrow \psi|_\nu = \texttt{true}$ if and only if $|\varphi|_\nu = \texttt{true}$ is equivalent to $|\psi|_\nu = \texttt{true}$.

About Formulas
Syntactic conventions and semantic properties

The following properties and conventions hold:

- $\vee$ and $\wedge$. **Commutative** w.r.t Tarski's semantics: $|\varphi \vee \psi|_\nu = |\psi \vee \varphi|_\nu$. **Associative** as well: $|\psi_1 \vee (\psi_2 \vee \psi_3)|_\nu = |(\psi_1 \vee \psi_2) \vee \psi_3|_\nu$.
- $\Rightarrow$ and $\Leftrightarrow$. By convention, **right associative**: $\psi_1 \Rightarrow \psi_2 \Rightarrow \psi_3$ means $\psi_1 \Rightarrow (\psi_2 \Rightarrow \psi_3)$.
- Priority rules. The order $\Leftrightarrow < \Rightarrow < \wedge < \vee < \neg$ applies by convention.

About Formulas

An example

$$(\neg X \lor Y \land Z \Rightarrow U) \Leftrightarrow V$$
$$((\neg X) \lor Y \land Z \Rightarrow U) \Leftrightarrow V$$
$$(((\neg X) \lor Y) \land Z \Rightarrow U) \Leftrightarrow V$$
$$((((\neg X) \lor Y) \land Z) \Rightarrow U) \Leftrightarrow V$$

About Formulas

Tautologies and antilogies

Tautology

A propositional formula φ such that for any valuation ν, $|\varphi|_\nu = \texttt{true}$.

Antilogy

A propositional formula φ such that for any valuation ν, $|\varphi|_\nu = \texttt{false}$.

Satisfiable

A propositional formula φ such that there exists a valuation ν verifying $|\varphi|_\nu = \texttt{true}$.

Semantic Equivalence

A proper definition

Equivalence

φ and ψ are **semantically equivalent** if for any valuation ν, $|\varphi|_\nu = |\psi|_\nu$. Then $\varphi \equiv \psi$.

Any tautology is semantically equivalent to $\top$, and any antilogy to $\bot$.

Semantic Equivalence

An equivalence relation

The semantic equivalence $\equiv$ is an equivalence relation:

- **Reflexive.** $\varphi \equiv \varphi$.
- **Symmetric.** $\varphi \equiv \psi$ if and only if $\psi \equiv \varphi$.
- **Transitive.** If $\psi_1 \equiv \psi_2$ and $\psi_2 \equiv \psi_3$ then $\psi_1 \equiv \psi_3$.

A property of sub-formulas

Let ψ_1 be a **sub-formula** of φ_1. If $\psi_2 \in \mathcal{F}_0$ is such that $\psi_1 \equiv \psi_2$, then replacing ψ_1 with ψ_2 in φ_1's definition results in a new formula $\varphi_2 \in \mathcal{F}_0$ such that $\varphi_1 \equiv \varphi_2$.

It can be proven by structural induction on φ.

Property

$\varphi \equiv \psi$ if and only if $(\varphi \Leftrightarrow \psi)$ is a tautology.

It's a consequence of Tarski's semantics.

Formal Logics

Properties of Propositional Formulas

Adrien Pommellet

March 27, 2025

Truth table of φ

A table that sets out the **values** of $|\varphi|_\nu$ for **each possible valuation** ν of its relevant logical variables.

Conventionally, we write `true` $:= 1$ and `false` $:= 0$ in truth tables.

Truth Tables
The main operators I

A	B	$A \wedge B$
0	0	0
0	1	0
1	0	0
1	1	1

A	B	$A \vee B$
0	0	0
0	1	1
1	0	1
1	1	1

A	B	$A \Rightarrow B$
0	0	1
0	1	1
1	0	0
1	1	1

Truth Tables
The main operators II

A	B	$A \Leftrightarrow B$
0	0	1
0	1	0
1	0	0
1	1	1

A	$\neg A$
0	1
1	0

Truth Tables
An example

Prove that $\psi = P \Rightarrow Q \Rightarrow P$ is a tautology.

E

P	Q	$Q \Rightarrow P$	$P \Rightarrow (Q \Rightarrow P)$
0	0	1	1
0	1	0	1
1	0	1	1
1	1	1	1

Practical Application

Exercise 1. Prove that $\varphi = A \vee B \Rightarrow (A \Rightarrow C) \Rightarrow (B \Rightarrow C) \Rightarrow C$ is a tautology.

Answer

Truth Tables

Equivalence

Property

Two formulas with the same set of input variables are equivalent if and only if they have the same truth table.

It is a direct consequence of the definition of truth tables.

Properties of $\mathcal{F}_0$

Distributivity and De Morgan's laws

Distributivity

For any $P, Q, R \in \mathcal{F}_0$:

$$P \vee (Q \wedge R) \equiv (P \vee Q) \wedge (P \vee R)$$
$$P \wedge (Q \vee R) \equiv (P \wedge Q) \vee (P \wedge R)$$

De Morgan's laws

For any $P, Q \in \mathcal{F}_0$:

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

Properties of $\mathcal{F}_0$

Double negation and material implication

Double negation

For any $P \in \mathcal{F}_0$:

$$\neg(\neg P) \equiv P$$

Material implication

For any $P, Q \in \mathcal{F}_0$:

$$(P \Rightarrow Q) \equiv (\neg P \vee Q)$$

Properties of $\mathcal{F}_0$

Double implication and law of the excluded middle

Double implication

For any $P, Q \in \mathcal{F}_0$:
$$(P \Leftrightarrow Q) \equiv (P \Rightarrow Q) \wedge (Q \Rightarrow P)$$

Law of the excluded middle

For any $P \in \mathcal{F}_0$, $P \vee \neg P$ is a **tautology** and $P \wedge \neg P$ is an **antilogy**.

Properties of $\mathcal{F}_0$

Simplifying formulas

As a consequence of the previous properties:

Theorem

Given a formula $\varphi \in \mathcal{F}_0$, there exists $\psi \in \mathcal{F}_{\{\perp,\neg,\wedge,\vee,\Rightarrow\}}$ such that $\varphi \equiv \psi$.

We can therefore **rewrite** formulas (here, by replacing $\Leftrightarrow$).

Practical Application

Exercise 2. You've just met three people named Alice, Bob, and Carl. They make the following statements:

- Alice. "Exactly one of us is telling the truth."
- Bob. "We are all lying."
- Carl. "The other two are lying."

Can you determine who's lying, and who's telling the truth?

Answer I

Answer II

Optional Homework

- Prove two of the aforementioned properties using truth tables.
- Exercises **1A** and **1B** of the 2019-2020 exam.

Formal Logics

The Satisfiability Problem

Adrien Pommellet

March 27, 2025

Introducing SAT

A definition

SAT

The **satisfiafiability problem** (also written `SAT`) consists in determining whether a formula $\varphi \in \mathcal{F}_0$ is satisfiable, that is, whether there exists a valuation ν such that $|\varphi|_\nu = \texttt{true}$.

Programs meant to solve this problem are known as `SAT` **solvers**.

Introducing SAT

Why SAT solvers?

SAT solvers can actually be used to solve a wide variety of problems.

Practical Application

Exercise 1. A, B, C, D, and E all live in a house together. We want to find who is at home and who isn't.

1. If A is at home then so is B.
2. D, E, or both are at home.
3. Either B or C, but not both, are at home.
4. D and C are either both at home or both not at home.
5. If E is at home then A and D are also at home.

Express this problem as a SAT instance.

Answer I

Variables

Answer II

Constraints

Answer III

A SAT instance

Answer IV

Looking for multiple solutions

Practical Application

Exercise 2. Can a generic graph $\mathcal{G} = (V, E)$ be coloured using a set C of 3 colours in such a manner two neighbouring vertices in V do not share the same colour? Express this problem as a SAT instance.

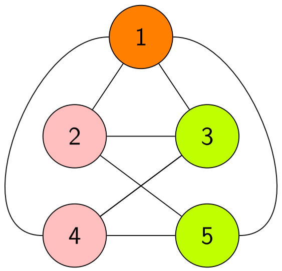


Figure 1: A well-coloured graph $\mathcal{G}$.

Answer I

Not so trivial variables

Answer II

A constraint on the encoding

Answer III

The original problem as a constraint

Properties of SAT

NP-completeness

Theorem (Cook's)

SAT is NP-complete.

Intuitively, a problem $\mathcal{P}$ is NP if it is **easy to check** (in polynomial time) whether an answer is valid or not.

It **may** still be hard to find a solution (brute forcing SAT is exponential).

Such a problem $\mathcal{P}$ is also NP-complete if any instance of another NP problem can easily (in polynomial time) be **reduced** to an instance of $\mathcal{P}$.

Properties of SAT

Negative normal form

Negative normal form

A formula $\varphi \in \mathcal{F}_0$ is said to be in negative normal form (NNF) if:

- The only constructors connecting sub-statements of φ are $\vee$ and $\wedge$.
- The $\neg$ constructor only appears in front of atomic statements.

Theorem

Given a formula $\varphi \in \mathcal{F}_0$, there exists $\psi \in \mathcal{F}_0$ in NNF such that $\varphi \equiv \psi$.

Consider a proof by **induction** using De Morgan's laws, double negation, material implication, and double implication.