CHAPITRE 1

Propositional Logic

1.1. Introduction

1.1.1. What is a proof?

DÉFINITION. A *proof* is a piece of text written by a human to convince another human that some fact is true.

PROBLEM 1.1.1. [**?**, p. 2]

You have been selected to serve on jury duty for a criminal case. The attorney for the defense argues as follow :

"If my client is guilty, then the knife was in the drawer. Either the knife was not in the drawer, or Jason Pritchard saw the knife. If the knife was not there on October 10, it follows that Jason Pritchard did not see the knife. Furthermore, if the knife was there on October 10, then the knife was in the drawer and also the hammer was in the barn. But we all know that the hammer was not in the barn. Therefore, ladies and gentlemen of the jury, my client is innocent."

Are you convinced ?

Problems :

- What is the goal ?
- What are the hypothesis ?
- What is the logical link between the statements ?
- There is plenty of unrelevant information.

A good proof consists of :

- A description of the goal
- Hypothesis
- Statements, one after the other.
  Each one derives logically from the previous ones.
- Conclusion

Let's look at a very simple example :

THÉORÈME. *Let x and y be two integers.*
If x is odd and y is odd, then xy is odd.

DÉMONSTRATION. Assume x is odd and y is odd.
Let k be an integer such that $x = 2k + 1$.
Let l be an integer such that $y = 2l + 1$.
Then, $xy = (2k + 1)(2l + 1) = 2(2kl + k + l) + 1$.
Note that $2kl + k + l$ is an integer.
Conclusion : xy is odd. □

1.1.2. Formal and Informal proofs. What do we mean by "derives logically from the previous ones" ?

What operations shall we consider as legal ?

To learn more about this, we need a *model* of what we intuitively consider as true, or *logic*.

This model we are going to construct is called *formal logic*.

1.1.2.1. *Formal systems.*

Définition. A *formal system* is build by providing :
– a set of strings : the *axioms*
– a set of rewriting rules

A string that can be obtained from the axioms by applying rewriting rules is called a *theorem* of the formal system.

The *proof* of a theorem is a description of the axioms and rules used to get the theorem.

The set of all theorems is called the *language* defined by the formal system.

Exemple. The MIU formal system[**?**]

A formal system does not need to have any *meaning* or *semantic*. It's pure syntax.

We will build formal logic as a formal system, with a meaning !

– The strings will be logic formulas
– The axioms will be simple logic formulas that we interpret intuitively true
– The rewriting rules will be transformations we interpret intuitively as valid

Then, any theorem of this formal system shall be interpreted as true.

We will have to be very careful with the choice of the axioms and rules !

1.1.2.2. *Proofs and programming.* Writing a proof and a writing a program are very similar :

Proofs	Programming	Purpose
Theorem	Function prototype	How to use it
Proof	Function definition	How it works
Intuitive proof	Comments	Explanation for the reader
Informal proof	Program in a high level language	Human readable
Formal proof	Program in assembly language	Machine readable
Checking an informal proof	Compiling	Human readable -> Machine readable

This comparison is not gratuitous. There are deep links between proofs and programming which have been studied a lot in the last decades. This is called the Curry-Howard correspondence.

1.2. Statements, Symbolic Representation, and Tautologies

As a first step, let's define logic formulas and their interpretation.

1.2.1. Statements and Propositions.

DÉFINITION. A *statement* (or *proposition*) is a sentence which is either true or false.

EXEMPLE. Which sentences below are statements ?
– The sky is cloudy
– It's raining
– Are you happy ?
– He is happy
– Three divides five
– There are life forms on other planets in the universe

NOTATION 1.2.1. A, B, C : symbols representing a statement.

Each of those symbols can be given a truth value, either True (T for short) or False (F for short).

1 is a statement if truth value True.

0 is a statement if truth value False.

1.2.2. Connectives.
How can we combine two statements ?

1.2.2.1. *And (conjunction), or (disjunction).*

EXEMPLE. "The sky is cloudy" : A. "It's raining" : B.
– "The sky is cloudy AND it's raining" : $A \wedge B$
– "The sky is cloudy OR it's raining" : $A \vee B$

DÉFINITION. The *truth table* of a formula describes when it's true or not, given the truth value of the symbols.

A	B	$A \wedge B$	$A \vee B$
T	T		
T	F		
F	T		
F	F		

1.2.2.2. *Exclusive or.*

EXEMPLE. "Cheese or desert"

You get cheese : A

You get desert : B

DÉFINITION. A exclusive or B : A xor B.

Truth table :

A	B	A xor B
T	T	
T	F	
F	T	
F	F	

1.2.2.3. *Implication.*

EXEMPLE. “If I get my driver license, I will open a bottle of champagne”.

“I get my driver license” : A.

“I open a bottle of champagne” : B.

DÉFINITION. A *implies* B : $A \to B$.

Truth table :

A	B	$A \to B$
T	T	
T	F	
F	T	
F	F	

Why this should be true :

- no real meaning but we have to pickup one to make it easy
- Most reasonable is true

Why this should be false :

This one can be tricky. To help you make up your mind, here are some examples :

- If I don’t get my driver license, but still I open a bottle of champagne, am I a liar ?
- “If I open a bottle of champagne, I will get my driver license”
- “When pig flies, I’ll give you 2 billions dollars”

Well, as strange as it can sound, $F \to T$ is a true formula.

If you start from an utterly false hypothesis, you can prove anything.

“Avec des si, on pourrait mettre Paris en bouteille”

“With ifs, Paris could be put in a bottle”

REMARQUE. $A \to B$ and $B \to A$ are not the same !

Sounds stupid ?

Look carefully around you ! This is the most common logic error in peoples argumentations.

And it’s not always an involuntary error.

EXERCICE 1. Give the antecedent and the consequent in the following sentences

- If the rain continues, then the river will flood
- A sufficient condition for network failure is that the central switch goes down
- It’s raining only if there are clouds in the sky
- It’s raining if there are clouds in the sky
- It’s raining if and only if there are clouds in the sky
- The avocados are ripe only if they are dark and soft
- A good diet is a necessary condition for a healthy cat

1.2.2.4. *Equivalence.*

EXEMPLE. The knight will win if and only if his armor is strong

The knight will win : A.

His armor is strong : B.

This sentence contains two informations :
- The knight will win if his armor is strong : $A \leftarrow B$.
- The knight will win only if his armor is strong : $A \rightarrow B$.

DÉFINITION. *A is equivalent to B* : $A \leftrightarrow B$.

Truth table :

A	B	$A \leftrightarrow B$
T	T	
T	F	
F	T	
F	F	

1.2.2.5. *Not.* So far we defined binary connectives, that is connectives which took two arguments.

Here is now a unary connective, taking just one argument.

EXEMPLE. It will not rain tomorrow

The knight will win : A.

His armor is strong : B.

This sentence contains two informations :
- The knight will win if his armor is strong : $A \leftarrow B$.
- The knight will win only if his armor is strong : $A \rightarrow B$.

DÉFINITION. not A : A' (more usual notation : $\neg A$)

Truth table :

A	A'
T	
F	

1.2.3. Complex Well Formed Formulas.

DÉFINITION. A *formula* is a string of
- symbols : A, B, ...
- connectives : $\vee$, $\wedge$, $\rightarrow$, $\leftarrow$, $'$, ...
- parenthesis : (,), [,]

A *well formed formula* (or wff) is a formula which satisfies certain syntax rules.

NOTATION 1.2.2. Symbols representing wff : P, Q, R.

1.2.3.1. *Order of precedence.* The formula $A \wedge B \rightarrow C'$ is ambiguous. It can be read as :
– $(A \wedge B) \rightarrow (C')$,
– $A \wedge [B \rightarrow (C')]$,
– $[(A \wedge B) \rightarrow C]'$

Solution 1 : always put all parenthesis (not very readable).

Solution 2 : put an order of precedence between the operators :

(1) Connectives within parentheses, innermost parenthesis first
(2) $'$
(3) $\wedge$, $\vee$
(4) $\rightarrow$, $\leftarrow$
(5) $\leftrightarrow$

With this order of preference, $A \wedge B \rightarrow C'$ unambiguously refer to $(A \wedge B) \rightarrow (C)'$.

1.2.3.2. *Evaluation of a wff.* From the truth values of the basic statements, and the elementary truth tables, one can *evaluate* the truth value of a wff.

Exemple. $A \wedge B \rightarrow C'$, with A true, B false and C true :

$A \wedge B$: false

C' : false

$A \wedge B \rightarrow C'$: true

There is a recursive algorithm to compute the evaluation of a wff.

1.2.3.3. *Truth table of a wff.* This yields an algorithm for computing the truth table of a wff.

A	B	C	$A \wedge B$	C'	$(A \wedge B) \rightarrow (C')$
T	T	T			
T	T	F			
T	F	T			
T	F	F			
F	T	T			
F	T	F			
F	F	T			
F	F	F			

Problem 1.2.3. What's the size of a truth table ?

Problem 1.2.4. Can any truth table be obtained by a wff ?

1.2.4. Tautologies and Contradictions.

Définition. A *tautology* is a statement that is always true.

A c*ontradiction* is a statement that is always false.

Exemple. $A \vee A'$

"Today the sun shine or today the sun does not shine".

EXERCICE 2. Are the following wff tautologies or contradictions ?

(1) $A \wedge A'$
(2) $A \leftrightarrow A$
(3) $(A \wedge B) \leftrightarrow (B \wedge A)$
(4) $(A \wedge B) \rightarrow A$

Algorithm to check for a tautology ?

Compute the truth table !

It's slow (2^n, where n is the number of propositions), but it works.

For certain simple forms of propositions, there is a much faster algorithm (see [**?**, Tautology test, p. 13]).

EXEMPLE. Is $[(A \wedge B \rightarrow C') \wedge (E \leftarrow D)] \leftrightarrow [(E \leftarrow D) \wedge (A \wedge B \rightarrow C')]$ a tautology ?

Let $P := (A \wedge B \rightarrow C')$ and $Q := (E \leftarrow D)$.

The formula above is $[P \wedge Q] \leftrightarrow [Q \wedge P]$.

Let's prove it's a tautology.

Take some truth values for A, B, C, D, E.

P and Q will then have some truth value.

But we know that $(A \wedge B) \leftrightarrow (B \wedge A)$ is a tautology.

So $[P \wedge Q] \leftrightarrow [Q \wedge P]$ will be true !

We don't actually need to compute the truth table of P and Q.

REMARQUE. Whenever you have some tautology, if you replace ALL occurrences of some basic statement (say A) by a more complicated wff, you still get a tautology.

1.2.5. Equivalence of wff. Let P and Q be two wff, and suppose $P \leftrightarrow Q$ is a tautology.

Then P and Q have the same truth table.

DÉFINITION. P and Q are *equivalent* (noted $P \Leftrightarrow Q$) if they have the same truth table.

REMARQUE. Writing $P \leftrightarrow Q$ or $P \Leftrightarrow Q$ is different.

In the first case, you just represent a formula, which can be true or false.

In the second case, you claim that the wff $P \leftrightarrow Q$ is a tautology.

1.2.5.1. *Basic equivalent wff.* What is the negation of “Peter is tall and thin” ?

Identity (id)	$A \vee 0 \Leftrightarrow A$ $A \wedge 1 \Leftrightarrow A$
Complement (comp)	$A \vee A' \Leftrightarrow 1$ $A \wedge A' \Leftrightarrow 0$
Commutativity (comm)	$A \vee B \Leftrightarrow B \vee A$ $A \wedge B \Leftrightarrow B \wedge A$
Associativity (ass)	$(A \vee B) \vee C \Leftrightarrow A \vee (B \vee C) \Leftrightarrow A \vee B \vee C$ $(A \wedge B) \wedge C \Leftrightarrow A \wedge (B \wedge C) \Leftrightarrow A \wedge B \wedge C$
Distributivity (dist)	$A \vee (B \wedge C) \Leftrightarrow (A \vee B) \wedge (A \vee C)$ $A \wedge (B \vee C) \Leftrightarrow (A \wedge B) \vee (A \wedge C)$
Double negation (dn)	$A'' \Leftrightarrow A$
De Morgan's Law (dm)	$(A \vee B)' \Leftrightarrow A' \wedge B'$ $(A \wedge B)' \Leftrightarrow A' \vee B'$
Implication (imp)	$A' \vee B \Leftrightarrow A \rightarrow B$

Remarque. Notice the duality between $\vee$ and $\wedge$.

Unlike $+$ and $\cdot$ they play a perfectly symmetric role in all formulas.

1.2.5.2. *More equivalent wff using substitution.*

Exemple. $(A \wedge B) \wedge 1 \Leftrightarrow (A \wedge B)$.

Remarque. Let P and Q be equivalent wff.

If you replace all occurrences of some basic statement by another wff, you still get two equivalent wff's.

Exemple. $A \wedge (B \vee C) \Leftrightarrow A \wedge (C \vee B)$

Remarque. Let P and Q be equivalent wff.

If P appears *exactly* as a subformula of a bigger wff, then you can substitute P by Q and get an equivalent wff.

1.2.5.3. *What's the point ?* Rewriting a formula to simplify it :

Exercice 3. [**?**, Example 7 p.12]

```
If ((outflow > inflow) and not ((outflow>inflow) and (pressure < 1000)))
  Do something;
Else
  Do something else;
EndIf
```

Write the wwf corresponding to the test.

Rewrite this wwf using equivalence rules.

Electronic circuits and logical gates :

Exercice 4. Build an electronic circuit for $(A \wedge B) \rightarrow C'$.
– Case 1 : you have electronic gates TRUE, FALSE, AND, OR, NAND.
– Case 2 : you only have electronic gates NAND.

1.3. Propositional logic

1.3.1. Arguments. We have seen propositions, well formed formula for formalizing statements.

We now need to formalize arguments.

EXEMPLE. [**?**, Example 17 p. 28]

Let's formalize the following argument :

"If interest rates drop, the housing market will improve.

Either the federal discount rate will drop, or the housing market will not improve.

Interest rates will drop.

Therefore the federal discount rate will drop."

Basic statements :
- I : Interest rates drop
- H : The housing market will improve
- F : The federal discount rate will drop

Argument : $(I \to H) \wedge (F \vee H') \wedge I \ \to \ F$

DÉFINITION. An *argument* is a wff of the form $P_1 \wedge P_2 \wedge \cdots \wedge P_n \ \to \ Q$.

$P_1, \cdots, P_n$ are the *hypothesis*.

Q is the *conclusion*.

The argument is *valid* iff $P_1 \wedge P_2 \wedge \cdots \wedge P_n \ \to \ Q$ is a tautology.

EXERCICE 5. Prove that the following arguments are valid :

(1) $[A \to (B \to C)] \ \to \ [(A \wedge B) \to C]$
(2) $[(A \wedge B) \to C] \ \to \ [A \to (B \to C)]$

Why is this reasonable ?

"If the sun shine then if you don't put sun cream, you will get burned" : $A \ \to \ (B \to C)$

"If the sun shine and you don't put sun cream, then you will get burned" : $(A \wedge B) \ \to \ C$

EXEMPLE. "George Washington was the first president of the United States.

Therefore everyday has 24 hours."

We put some more restrictions to only keep "meaningful" arguments.
- Q has some relation with the P_i.
- the P_i are not in contradiction

1.3.2. Proof sequences. So now, how to prove that an argument is a tautology ?

(1) Use the truth table
(2) Use formal logic !

EXEMPLE. Let's prove that the argument $A \vee (A \vee B)' \ \rightarrow \ (A \vee B')$ is valid.

We are lazy to compute the truth table.

Instead, we assume that the hypothesis is valid, and use a series of equivalences to deduce that the conclusion is valid.

$$\begin{array}{lll} 1. & A \vee (A \vee B)' & (\text{Hypothesis}) \\ 2. & A \vee (A' \wedge B') & (\text{De Morgan}'s,\ 1) \\ 3. & (A \vee A') \wedge (A \vee B') & (\text{Distributivity},\ 2) \\ 4. & 1 \wedge (A \vee B') & (\text{Complement},\ 3) \\ 5. & (A \vee B') & (\text{Identity},\ 4) \end{array}$$

What we just did is called a *proof sequence.*

This is our first proof using *formal logic.*

EXERCICE 6. Prove that the following argument is valid :
$(A \vee B)' \vee (A' \wedge B) \ \rightarrow \ A'$

DÉFINITION. *Formal Logic* : wff + set of derivation rules (formal system).

What derivation rules shall we accept ?

We want our formal logic to be :
- Correct
- Complete
- Minimal
- Powerful

1.3.3. Looking for derivation Rules.

EXEMPLE. $A \wedge B \ \rightarrow \ A$

Equivalences rules are not enough.

We can't prove the previous statement, because $A \wedge B$ is strictly stronger than A.

We have build equivalence rules from equivalences $P \leftrightarrow Q$.

We can build *Inference rules* from any valid argument $P \rightarrow Q$.

Rule name	From	Can derive
modus ponens (mp)	P, $(P \rightarrow Q)$	Q
modus tollens (mt)	$(P \rightarrow Q)$, Q'	P'
conjunction (con)	P, Q	$P \wedge Q$
simplification (sim)	$P \wedge Q$	P, Q
addition (add)	P	$P \vee Q$

EXEMPLE. $A \wedge (A \rightarrow B) \wedge (B \rightarrow C) \ \rightarrow \ C$

$$\begin{array}{lll} 1. & A \wedge (A \rightarrow B) \wedge (B \rightarrow C) & (\text{Hypothesis}) \\ 2. & A & (\text{Simplification},\ 1) \\ 3. & A \rightarrow B & (\text{Simplification},\ 1) \\ 4. & B & (\text{Modus ponens},\ 2,\ 3) \\ 5. & B \rightarrow C & (\text{Simplification},\ 1) \\ 6. & C & (\text{Modus ponens},\ 4,\ 5) \end{array}$$

EXEMPLE. $(A \to (B \vee C)) \wedge B' \wedge C' \ \to \ A'$

1.	$(A \to (B \vee C)) \wedge B' \wedge C'$	(Hypothesis)
2.	$(A \to (B \vee C))$	(Simplification 1)
3.	$(B' \wedge C')$	(Simplification 1)
4.	$(B \vee C)'$	(De Morgan's 3)
5.	A'	(Modus tollens 4, 5)

EXERCICE 7. Prove the validity of the following arguments using formal logic :

(1) $A \wedge (B \to C) \wedge [(A \wedge B) \to (D \vee C')] \wedge B \to D$

(2) $A' \wedge (A \vee B) \to B$

(3) $A' \wedge B \wedge [B \to (A \vee C)] \to C$

(4) $A \wedge A' \to B$

THEOREM 1.3.1. *This formal logic is correct and complete.*

The correctness of the formal logic comes from the way we built the rules from valid arguments.
Proving it's complete is far more difficult.

1.3.4. Methods to make it easier. Our logic is complete, but still not very practical to use.
Here are some tricks to make it easier.
Most of the time, the proofs start like this :

1.	$(A \to (B \vee C)) \wedge B' \wedge C'$	(hyp)
2.	$(A \to (B \vee C))$	(Simp 1)
3.	$(B' \wedge C')$	(Simp 1)
...	...	...

The steps 2 and 3 are pretty trivial, and it's reasonable to start directly with :

1.	$(A \to (B \vee C))$	(hyp)
2.	$(B' \wedge C')$	(hyp)
...	...	...

1.3.4.1. *The deduction method.*

EXEMPLE. $(A \to B) \wedge (B \to C) \ \to \ (A \to C)$
The obvious start is :

1.	$(A \to B)$	(hyp)
2.	$(B \to C)$	(hyp)

But then it's pretty painful :

3.	$A' \vee B$	(imp 1)
4.	$A' \vee (B \to C)$	(add 2)
5.	$(A' \vee B) \wedge (A' \vee (B \to C))$	(con 3, 4)
6.	$A' \vee (B \wedge (B' \to C))$	(dist 5)
7.	$A' \vee C$	(mp 6)
8.	$A \to C$	(imp 7)

Here it's easier to first rewrite the argument into an equivalent argument.
We have seen that $P \to (Q \to R)$ and $(P \wedge Q) \to R$ are equivalent.
So we can prove $A \wedge (A \to B) \wedge (B \to C) \;\to\; C$ instead, which is straightforward :

1.	A	(hyp)
1.	$(A \to B)$	(hyp)
2.	$(B \to C)$	(hyp)
3.	B	(mt 1, 3)
4.	C	(mt 2, 4)

This trick is called the *deduction method.*
It's powerful because we have :
- a simpler goal
- more hypothesis

Exercice 8. $(A \to B) \wedge [B \to (C \to D)] \wedge [A \to (B \to C)] \;\to\; (A \to D)$

1.3.4.2. *Additional deduction rules.* As we have seen, any valid argument could be transformed into a new rule.

Exemple. Here are two useful rules.

Rule name	From	Can derive
hypothetical syllogism (hs)	$P \to Q,\, Q \to R$	$P \to R$
contraposition	$P \to Q$	$Q' \to P'$

Adding those new rules won't change the power of the formal logic, but can make it more convenient. On the other hand, a minimal formal logic is nice since it's less error prone, and it's easier to remember all the rules.
It's up to you to choose which rules you want to use (and learn).

Exemple. $(P \to Q) \wedge (P' \to Q) \to Q$.

1.4. Verbal arguments

We can now start checking verbal arguments.

Exemple. Let's prove the following verbal argument[**?**, Example 17 p. 28] :
"If interest rates drop, the housing market will improve.
Either the federal discount rate will drop, or the housing market will not improve.
Interest rates will drop.
Therefore the federal discount rate will drop."

Exercice 9. Prove the following verbal argument[**?**, Ex. 30 p. 32] :
"If the program is efficient, it executes quickly.
Either the program is efficient or it has a bug.
However, the program does not execute quickly.
Therefore it has a bug."

1.5. Summary

1.5.1. Different levels of logic. Three levels for arguments :

– formal argument as a string of symbols, proof sequence (syntactic level).
– formal argument as a truth table (semantic level).
– verbal argument (interpretation level).

Any verbal argument based on propositions can be translated into a formal argument.

The propositional logic system is complete and correct :

– Any formal argument proved syntactically is valid semantically.
– Any semantically valid formal argument can be proved syntactically (it can be hard, though !).

1.5.2. Syntactic proof versus semantic proof. Testing an argument by computing it's truth table is purely mechanical.

So, what's the point of formal logic ?

– It's often faster and shorter to write
– You have more chance to grab the meaning of an argument
– In predicate logic we won't be able to rely on truth tables