# Mathematical Logic

## Massimo Perego

## Contents

# Introduction

Starting on logic: from the sentences

- Every man is mortal
- Socrates is a man

-------------------------

- Therefore, Socrates is mortal

Under the line logical conclusions.

This is an example of **syllogism**. **Consequences from some assumptions, which are considered true**.
The concept is that **if the assumptions are true, then the consequences must also be true**.

Logic doesn't really think about each part of the sentence in a literal sense (we don't care what they "mean" in the real world), but **each sentence has to be formalized in a way that makes the "shape" of the information the most important part**.

Another example:

- All cats have seven legs
- Bob is a cat

-------------------------

- Therefore Bob has seven legs

We can say this because the "shape" of the argument is the same as before, even if rationally it doesn't make sense, it's not "right", but **in a world in which these informations are true then the consequences must be true**.

We don't care about the content, **we care about the shape and what our clauses imply**.

Another example:

- Every tik is tok
- Tak is tik

_____

- Therefore tak is tok

It doesn't mean shit, but this is true whatever being tik, tak or tok means.

We probably will want to model pieces of the real world, but logic doesn't care about that.

**Formalizing** the shape of these examples:

- First sentence
$$\forall x \quad P(x) \to Q(x)$$

  For all individual in the world if it has the property of being tik($P()$), then it has the property of being tok ($Q()$).
- Second sentence
$$P(s)$$

  Tak ($s$) is tik

_____

- Therefore
$$Q(s)$$

  Tak is tok

We **formalized** the sentences, but we **lost some information in the process**, we lose the facts that we're talking about Socrates, or cats, we don't know that we're talking about mortality or number of legs.

Considering another argument:

- Every flower is perfumed
- The rose is perfumed

---

- Therefore the rose is a flower?

Formalization

$$\forall x \quad P(x) \rightarrow Q(x)$$

$$Q(s)$$

---

Therefore $P(s)$?

This **doesn't hold**, our first predicate doesn't automatically works bidirectionally, we're **reversing our implication**.
With our assumptions there's no "proof" that $P(s)$ is valid if $P(x) \rightarrow Q(x)$.


Final example:

- If it rains I take my umbrella

Which one is the **equivalent?**

1. If it doesn't rain I don't take my umbrella
2. If I don't take my umbrella then it doesn't rain
3. If I take the umbrella then it rains
4. Either it doesn't rain or I take the umbrella
5. It rains only if I take the umbrella
6. It rains if I take the umbrella
7. It rains if and only if I take the umbrella
8. None of the above

To answer we need to formalize what we mean by equivalent: for logicians it means "given a world in which its true then the other is true and given a world in which it's false then the other is false".

With this definition then the **second** one is **true**:

$$P(x) \rightarrow Q(x) \text{ then } \neg Q(x) \rightarrow \neg P(x)$$

The first one is a simple negation, it can't be true as well.

The third one is the opposite of the original sentence.

There's at least two types of *or* in a natural language, usually it's exclusive. The **fourth sentence is correct if the *or* is inclusive**, if it's a *xor* it can't be.

**Five** is also a **correct** translation, once we agree on the meaning of words like "if", "only if", ...; It basically means that the first part (rain) can or can not happen based on the "umbrella" fact, which is the same as the original. We're speaking in both cases about a necessary and sufficient condition for rain. Taking the umbrella is a necessary condition for rain, it doesn't cause the rain. There's no causation between clauses.

Six and seven are false, since five is true.

**Equivalence:** What does "saying the same thing" means in logic terms? Two sentences **say the same thing if and only if (iff) they are true in exactly the same circumstances**, a.k.a. "possible world", a.k.a. "truth evaluation".

If *it rains* then *I take my umbrella*: the cursive ones are stand alone sentences, each one is an **atomic sentence** (in logic terms), they cannot be further disassembled.

So, **formalizing** the sentence

- it rains $= P$
- I take my umbrella $= Q$
- if/then $= \rightarrow$

$$P \rightarrow Q$$

There can be many (infinite) world in which "it rains", i.e. $P$, is true and another set of worlds in which "I take my umbrella", i.e. $Q$, is true. These sets can (will) intersect.

Imagining something along the lines of a Venn diagram (I don't care enough to actually draw it), there will be the following regions:

- $P$: worlds in which it only rains
- $Q$: worlds in which I only take my umbrella
- $P \cap Q$: worlds in which it rains and I take my umbrella
- $\neg(P \vee Q)$: worlds in which it doesn't rain and I don't take my umbrella (outside all)

For each region we have

- it rains but I don't take my umbrella (inside the $P$ region only) $\implies$ for the worlds in here the preposition is false
- it doesn't rain but I take my umbrella ($Q$ region) $\implies$ for the worlds in here the preposition is true
- inside both regions, it rains and I take my umbrella $\implies$ for the worlds here the preposition is true
- outside both regions, doesn't rain, don't take my umbrella $\implies$ for the worlds here the preposition is true

Now we have a sort of "image" on which to evaluate if other sentences are equivalent. **If another preposition gives us the same image**, i.e. the same truth values in the same regions of the diagram, **it's equivalent to our original**.

Considering our earlier example and taking sentence #4:

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

- $\neg P$ is true when $P$ is not (duh)
- $P \vee Q$ is true inside the $P$, $Q$ and intersection region between them

So, combining these two gives us that, $(\neg P) \vee Q$ is true

- inside $Q$ region
- inside intersection between $P$ and $Q$
- outside everything

So it's the same as our initial example $P \rightarrow Q$, they give the same "picture", so they say the same thing, they are equivalent. **They are true in the same circumstances**. Each world is false (or true), i.e. has the same value, in both cases.


Another case, sentence #2:
$$\neg Q \rightarrow \neg P$$
gives the same picture, while sentence #1

$$\neg P \rightarrow \neg Q$$

does not (obviously, they're opposite to one another, they can't be both true).


Considering sentences #5

$$Q \leftarrow P \implies P \rightarrow Q$$

and #6
$$Q \rightarrow P$$

#6 can be rephrased in #6′ as: If I take my umbrella it rains. Doesn't sound too correct, does it?
While in sentence #5 "only if" goes the other way in respect to "if".

Basically:

- **if** goes **one way**
- **only if** goes the **other way**
- **if and only if (iff)** goes **both ways**

Inside

$$P \to Q : \textit{true}$$

It means that:

- $P$ is **sufficient** for $Q$ to be true (if there's $P$ there must be $Q$)
- $Q$ is **necessary** for $P$ to be true (there can't be $P$ without $Q$)

There can be no case in which $P$ (it rains) is true and $Q$ isn't (I take my umbrella).

When is our preposition true? "$P \to Q$ is true" can be read as:

"When $P$ is true then $Q$ is true"

or

"Each single time $P$ is true then $Q$ is true (in every possible world)"

# 1 Introduction to Formulas

A **"Sentence"** (or **"proposition"**) in natural language is a **grammatically correct sequence of words such that it makes sense to ask whether** (in the given circumstance) **it is true or false**.

For example, "It rains" is a sentence, and we can wonder if in our world it's true or not (pretend not to be able to look out the window).

"If it rains I stay at home" is another sentence, it can be true or not.

"What's the time?" is NOT a sentence, it has "the time" as the answer, we don't wonder if it's true or false. In general, questions are not sentences.

"$2 + 2 = 4$" is a sentence, in math language, but it's a sentence. We didn't define which language. "$2 + 2 = 5$" is also a sentence, usually wrong, but a sentence.

## 1.1 Meaning: Denotation vs Connotation

Considering:

- 4
- $2^2$
- $6 - 2$
- $5 + 1$

They're **not sentences** because they **can't be true or false**, they just are.

But all these expressions **mean the same thing**, they all **denote** the natural number 4 (apart from the last one). This is the denotation of these expressions.

They are all expressions that describe distinct ways to obtain 4. The **denotation** is the number 4, **all the other informations given by the expressions are connotations**.
For each given an expression $E$ we have the denotation of $E$ and the connotation(s) of $E$.

A **denotation** is just a **name** (a pointer), **referring to a uniquely determined object** of discourse.

The **connotations** are the **names with all the actual information they contain**. A name can be a complex thing and as such can contain informations.

We will generally be dealing with denotation, since it's hard to deal with connotation as they can be any way to deal with the object of interest.

**Invariance under substitutions:** denotation enjoys the following property: two different ways to connote the same denoted object can be substituted and they mean the same thing.

We can exchange $4$ and $2+2$, right? They are different strings but evaluate to the same object.

What is the denotation of sentences? What can we say about (considering our "usual" arithmetic world):

- $4 = pred(5)$ we can say that it's *true*
- $4 = succ(5)$: *false*
- $4 = 3 + 1$: *true*
- $4 = 4$: *true*
- $2 = 2$: *true*
- $4 = 6$: *false*

The **denotation of a sentence**, in a given possible world, is just **one of two truth values**, *true* or *false*. **Each sentence is evaluated to a precise truth value**.

**Tautology:** something that's true in every possible world, like $P(x) \lor \neg P(x)$ or $P \rightarrow P$. They can be indefinitely complex.

## 1.2 Formalization

What is propositional logic about? Propositions (a.k.a. sentences) and their semantics (denotation, their truth values) in every possible circumstance.

To formalize sentences we distinguish among **two types of sentences**:

- **Atomic sentences**, simple sentences (It rains, Paul runs, I stay at home, ...) which cannot be simplified anymore, can't be disassembled further (in propositional logic), there aren't any connectives.

- **Complex sentences**, made from atomic sentences with connectives in between (It rains and Paul runs, Paul runs or I stay at home, ...), that can be arbitrarily complex

The **formalization** of these types of sentences is:

- **Atomic sentences → atomic formulas**, they **become symbols** ($p$, $q$, $r$, $p_1$, ...), we fix an infinite (countable, don't need more) set of symbols, whatever they are, and they become a set of atomic formulas.

  $\mathcal{L}$ is called "propositional language" and its elements are called "atomic formulas" or "propositional letters/variables". In every possible world, every propositional letter can either be true or false. Basically: choose an infinite amount of symbols, each sentence is one.

- **Complex sentences**: fix a propositional language and the **set of propositional formulas on this language**.
  $\mathcal{F}_{\mathcal{L}}$ is defined as follows (3 ways):
  1. The smallest set such that:
     - for all $p \in \mathcal{L}$ then $p \in \mathcal{F}_{\mathcal{L}}(\mathcal{L} \subseteq \mathcal{F}_{\mathcal{L}})$
     - If $A, B \in \mathcal{F}_{\mathcal{L}}$ then $(A \cap B)$, $(A \cup B)$, $(A \to B)$, $(\neg A) \in \mathcal{F}_{\mathcal{L}}$
  2. $\mathcal{F}_{\mathcal{L}}$ is the intersection of all sets $\mathcal{X}$ such that:
     - $\mathcal{L} \subseteq \mathcal{X}$
     - If $A, B \in \mathcal{X}$ then $(A \cap B)$, $(A \cup B)$, $(A \to B)$, $(\neg A) \in \mathcal{X}$
  3. Inductive definition of $\mathcal{F}_{\mathcal{L}}$: $\mathcal{F}_{\mathcal{L}}$ is the set such that the following properties hold:
     - $\mathcal{L} \subseteq \mathcal{F}_{\mathcal{L}}$ (base)
     - If $A, B \in \mathcal{F}_{\mathcal{L}}$ then $(A \cap B)$, $(A \cup B)$, $(A \to B)$, $(\neg A) \in \mathcal{F}_{\mathcal{L}}$ (inductive step)
     - Nothing else belongs to $\mathcal{F}_{\mathcal{L}}$

The difference between first and last one is that in the former we specify "smallest set", while in the last it's a property.

Examples: $p, q, r \in \mathcal{L}$, then

$$((p \wedge q) \to r) \in \mathcal{F}_{\mathcal{L}}$$

this is a valid formula and as such is in $\mathcal{F}_{\mathcal{L}}$

$$p \vee \wedge (q \wedge r) \notin \mathcal{F}_{\mathcal{L}}$$

this is a string but it's not a formula.
We need to be able to recognize formulas from (ugly) strings. We need a certificate (a kind of proof) to certify that it's a formula.

It's worth noting that the set of connectives used in each definition is chosen arbitrarily, there can be indefinitely many (or few) and the formalization of complex sentences is based upon the considered connectives.

## 1.3 Certifying formulas

We need to **produce a certificate that a string is a formula**, a kind of proof that something is a formula. We don't need to prove that something is NOT a formula since we indirectly get that from the fact that there's no certificate for such formula.

### 1.3.1 $\mathcal{L}$-construction

First example of a certificate. An $\mathcal{L}$-construction is a way to **show that**, given **a word defined on the symbols in the alphabet** $w \in (\{\wedge, \vee, \neg, \rightarrow, ), (\} \cup \mathcal{L})^*$ ("w" is any strings made up of these symbols) **is actually a formula**, i.e. $w \in \mathcal{F}_\mathcal{L}$. Defines how to prove that $w$ is a formula.

We can consider $w$ as a sequence of strings $w_1, w_2, w_3, \ldots$ such that:

- $w_u = w$
- For all $i = 1, 2, \ldots, u$ one of these rules must hold
    - $w_i \in L$ ($i$th position is a letter)
    - or there is $j < i$ such that $w_i = (\neg w_j)$
    - or there are $j, k < i$ such that
        $$w_i = (w_j \wedge w_k)$$
        $$w_i = (w_j \vee w_k)$$
        $$w_i = (w_j \rightarrow w_k)$$

The string must be the negation of something else, the conjunction of something else or the disjunction of something else, implication, ecc.; it must be made of something else which is simpler.

An $\mathcal{L}$-construction is a sequence of finitely many strings such that the last one is the one we want to prove.

Example: for $(p \wedge q) \rightarrow r$ the sequence can be, adding one after the other:

$$p, q, p \wedge q, r$$

and using these we can build the original sentence, so we have an $\mathcal{L}$-construction for it and so $(p \wedge q) \rightarrow r \in \mathcal{F}_\mathcal{L}$.

This isn't the only $\mathcal{L}$-construction possible, there are infinitely many possible ones (we can add useless stuff and even repetitions), we just **need the elements to build our string and prove it's a formula**.

But we can have a **shortest** $\mathcal{L}$-construction, and at the minimum it will be as long as the number of letters + number of connectives used in the formula we need to certify.

Provide a definition for a minimal length $\mathcal{L}$-construction for a formula $w \in \mathcal{F}_{\mathcal{L}}$ and use it to show that $\rightarrow (p \wedge q) \notin \mathcal{F}_{\mathcal{L}}$.

$\mathcal{L}$-constructions should be (are) easy to find and easy to read.

### 1.3.2 Unique readability

With $\mathcal{L}$-constructions comes another property of the syntactical aspect of formulas: **unique readability**. From the definition of $\mathcal{L}$-construction, it can be observed that if $w \in \mathcal{F}_\mathcal{L}$ then **exactly one** of the following cases **holds**

- $w \in \mathcal{L}$
- or there is $v \in \mathcal{F}_\mathcal{L}$ such that $w = (\neg v)$
- or there are $v_1, v_2 \in \mathcal{F}_\mathcal{L}$ such that $w = (v_1 \wedge v_2)$
- or there are $v_1, v_2 \in \mathcal{F}_\mathcal{L}$ such that $w = (v_1 \vee v_2)$
- or there are $v_1, v_2 \in \mathcal{F}_\mathcal{L}$ such that $w = (v_1 \rightarrow v_2)$

Essentially, $w$ is not decomposable, or is decomposable in a unique way.

Example of a formal language lacking this property: let $w \in \Sigma^*$ and consider $w = v_1 \cdot v_2$ (concatenation of strings). This doesn't have unique readability, for example if we have $w = hello$ this is decomposable in $w = v_1 \cdot v_2$ in multiple ways

- $\epsilon, hello$
- $h, ello$
- $he, llo$
- $hel, lo$
- $hell, o$
- $hello, \epsilon$

It's decomposable 6 different ways, we just need 2 to prove that it doesn't have unique readability.

The language of formulas is designed to have unique readability.

Unique readability of formulas provides us with other kinds of certificates that prove something is a formula.

Unique readability provides us with a **unique parsing tree** for any formula.

We can decompose uniquely the formula: $(p \wedge q) \to r$ in a parse tree such that

$$(p \wedge q) \to r$$
$$\overset{\frown}{p \wedge q} \quad r$$
$$\overset{\frown}{p \quad q}$$
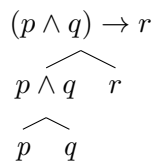
If we can have a single parsing tree it's a formula.

### 1.3.3 Structural induction principle

Since we can **define $\mathcal{F}_\mathcal{L}$ as an inductive set** (third definition), it comes with its own **induction principle**.

Let $\mathcal{P}$ be some property that can be asked of formulas (note that it may hold or not for any given formula, it doesn't have to be true, it's just a property for any given $\mathcal{F} \in \mathcal{F}_\mathcal{L}$).

This property $\mathcal{P}$ holds for all formulas $\mathcal{F} \in \mathcal{F}_\mathcal{L}$ iff:

1. **base**: $\mathcal{P}$ holds for all propositional letters $p \in \mathcal{L}$
2. **step(s)**: if $\mathcal{P}$ holds for generic formulas $A, B \in \mathcal{F}_\mathcal{L}$ then $\mathcal{P}$ holds for all the ways to build new formulas starting from $A$ and $B$: $(\neg A), (A \wedge B), (A \vee B), (A \rightarrow B)$

We can prove that this principle works since we defined $\mathcal{F}_\mathcal{L}$ in an inductive way. Proof of the structural induction principle:

- Let $\mathcal{P}$ be a property satisfying "base" and "step(s)" (our principle).
- Let $\mathcal{I} = \{\mathcal{F} \in \mathcal{F}_\mathcal{L} : \mathcal{P} \text{ holds for } \mathcal{F}\}$ (the set of formulas for which $\mathcal{P}$ holds).

We now need to shows that $\mathcal{F}_\mathcal{L} = \mathcal{I}$ (the set of formulas in which the property holds coincides with the complete set of formulas).

*Proof.* To do that we can show that:

- $\mathcal{F}_\mathcal{L} \subseteq \mathcal{I}$
- $\mathcal{I} \subseteq \mathcal{F}_\mathcal{L}$

The second one is trivial since $\mathcal{I}$ is defined as a subset of $\mathcal{F}_\mathcal{L}$. We only need to prove that all the formulas are in $\mathcal{I}$.

Now since $\mathcal{P}$ satisfies "base" then $\mathcal{P}$ holds for all $p \in \mathcal{L}$, for all propositional letters $p \in \mathcal{L}$, $p \in \mathcal{I}$ then $\mathcal{L} \subseteq \mathcal{I}$.
Since $\mathcal{P}$ satisfies "step(s)" then if $A, B \in \mathcal{I}$ then $\mathcal{P}$ holds for $A$ and $B$ and then by "step(s)" $\mathcal{P}$ holds for $(\neg A), (A \wedge B), (A \vee B), (A \rightarrow B)$, and thus the same constructions we defined for $\mathcal{F}_\mathcal{L}$ belong to $\mathcal{I}$. Each formula $\mathcal{F}$ then belongs to $\mathcal{I}$. This also means that $\mathcal{F}_\mathcal{L} \subseteq \mathcal{I}$.

$\square$

## 1.4 Semantics of propositional logic

The **meaning**, in the denotational sense, that **we want to attach to formulas**, which for sentences can be either true or false.

Classical propositional logic is based on a couple of **underlying design ideas**:

- **Bivalence principle**
- **Truth-functionality principle**

**Bivalence principle:** a sentence (formula) in every possible world **is either true or false**. Basically, we have just 2 truth values, either true or false.

**Truth-functionality principle:** (also called compositionality, extensionality) the **truth value of a sentence** (a formula) in a given possible world **only depends on the truth values of the sentences composing it** (and the fixed meaning of connectives).
We don't need to consider what the sentences says. We completely discard the meaning of the values, an atomic sentence is only either true or false and complex sentences with connectives lead to the logical meaning of the formula.

An example of a non truth-functional realm is probability since it's based on, well, probability, it's based on the likelihood of events, we can't connect events with truth-functionality, we must consider what each event means, we can't reason on probability without knowing what each probability means (which is what truth-functionality would imply).

Formalizing time?

For instance, take conjunction

$$A, B \in \mathcal{F}_{\mathcal{L}} \qquad A \wedge B$$

We want to evaluate the value in a given possible world.

By bivalence we know that this is either true or false.
By truth-functionality this value depends only on

- whether $A$ is true or false
- whether $B$ is true or false

**The notion of "possible world":** it's a **function** from the set of propositional letters to the set of truth values

$$v : \mathcal{L} \rightarrow \{0, 1\}$$

It **assigns a truth value to each of the propositional letters**. Basically it gives the truth value for every variable/letter.
A possible world (circumstance) is formalized by a map $v : \mathcal{L} \rightarrow \{0, 1\}$ called truth-evaluation or truth-assignment.

The **set of all possible worlds is the set of all possible truth evaluations**.

Semantics of $(A \wedge B)$: we can **extend** $v$ to apply it to things like $(A \wedge B)$ (basically evaluate the value of this):

$$\tilde{v}(A \wedge B) \in \{0, 1\}$$

Which depends only on $\tilde{v}(A)$ and $\tilde{v}(B)$ (the values of the atomic formulas that compose it) and on the fixed semantics of $\wedge$.

We want to combine the semantics of $\tilde{v}(A)$ and $\tilde{v}(B)$, fixing

$$I_{\wedge} : \{0, 1\}^2 \rightarrow \{0, 1\}$$

Basically the truth table of the connective $\wedge$.

$$\tilde{v}(A \land B) := I_\land(\tilde{v}(A), \tilde{v}(B))$$

$$I_\land : \{0,1\}^2 \to \{0,1\} = a \cdot b = \min\{a, b\}$$

| $a$ | $b$ | $a \land b$ |
|-----|-----|-------------|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

$$I_\lor : \{0,1\}^2 \to \{0,1\} = a + b = \max\{a, b\}$$

| $a$ | $b$ | $a \lor b$ |
|-----|-----|------------|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

$$I_\neg : \{0,1\} \to \{0,1\} = 1 - a$$

| $a$ | $\neg a$ |
|-----|----------|
| 0 | 1 |
| 1 | 0 |

$$I_\to : \{0,1\}^2 \to \{0,1\} = \max\{1 - a, b\}$$

| $a$ | $b$ | $a \to b$ |
|-----|-----|-----------|
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

Exercise: specify the function for co-implication $\leftrightarrow$ and exclusive or $\oplus$.

$$I_{\leftrightarrow} : \{0,1\}^2 \to \{0,1\} =$$

| $a$ | $b$ | $a \leftrightarrow b$ |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

Also, what's the relation between implication and another one of these functions?
tions?
Answer:

**Definition:**   A **truth evaluation is an arbitrarily chosen map**

$$v : \mathcal{L} \to \{0, 1\}$$

**Definition:**   the canonical extension $\tilde{v} : \mathcal{F}_{\mathcal{L}} \to \{0, 1\}$ of $v : \mathcal{L} \to \{0, 1\}$ is defined as follows:

- if $f = p$, $p \in \mathcal{L}$ then $\tilde{v}(p) = v(p)$
- if $A, B \in \mathcal{F}_{\mathcal{L}}$ then
  $$\tilde{v}(A \wedge B) := I_{\wedge}(\tilde{v}(A), \tilde{v}(B))$$
  $$\tilde{v}(A \vee B) := I_{\vee}(\tilde{v}(A), \tilde{v}(B))$$
  $$\tilde{v}(A \to B) := I_{\to}(\tilde{v}(A), \tilde{v}(B))$$
  $$\tilde{v}(\neg A) := I_{\neg}(\tilde{v}(A))$$

The canonical extension basically means that $v$ gives us the value of a single letter but can be extended for any valid formula.
The tilde might be forgotten.

### 1.4.1 Tautologies and other definitions

**Definition:** a **tautology** is a formula $F$ (so $F \in \mathcal{F}_\mathcal{L}$) is a tautology iff $\tilde{v}(F) = 1$ for all $v : \mathcal{L} \to \{0, 1\}$, i.e. it's always true.

**Definition:** a **satisfiable formula**, a formula $F \in \mathcal{F}_\mathcal{L}$ is satisfiable iff

$$\tilde{v}(F) = 1$$

for at least one assignment $v : \mathcal{L} \to \{0, 1\}$, i.e. there is at least one assignment that makes it true.

A tautology obviously is a satisfiable formula (implication).

**Definition:** a formula $F \in \mathcal{F}_\mathcal{L}$ is (these will be kinda synonyms in our usage)

- **contradiction**
- **unsatisfiable**
- **refutable**

iff $\tilde{v}(F) = 0$ for all assignments $v : \mathcal{L} \to \{0, 1\}$, i.e. is never true.

**Fact:** Let $F \in \mathcal{F}_\mathcal{L}$ then $F$ is a tautology iff

$$\neg F \text{ is a contradiction}$$

*Proof.* $F$ is a tautology iff (by definition of tautology) $\tilde{v}(F) = 1$ for all $v : \mathcal{L} \to \{0, 1\}$.

By definition of (truth table) $I_\neg \to$ iff $\tilde{v}(\neg F) = 0$ for all $v : \mathcal{L} \to \{0, 1\}$, and thus by definition of contradiction $\neg F$ is unsatisfiable. $\qquad\square$

To **prove** that the **implication is correct** the way it is:

*Proof.* I mean, not really a proof but kind of (not a proof, I just like the little square at the end). By referring to the truth table of

$$I_\rightarrow : \{0,1\}^2 \rightarrow \{0,1\}$$

As seen some pages behind. Exercise: write down all possible truth tables on 2 arguments and assign a fit name (or a formula) to each one of them and find the only one that can reasonably named implication $a \rightarrow b$.

For $n$ variables there are $2^{2^n}$ possible truth tables. To denote the all false table $\bot$, to denote all true $\top$.

I won't actually write all truth tables, I trust the guy.

Progression on combinatorics:

$$|\{F : \{0,1\}^n \rightarrow \{0,1\}\}| = 2^{2^n}$$

Which is a specific version for a more general formula: with $A$ and $B$ finite sets
$$|\{F : A \rightarrow B\}| = |B|^{|A|}$$

So $|B|$ options for each element of $A$.

Let $\mathcal{S}$ be a set of $n$ many elements. There are $2^n$ possible subsets of $\mathcal{S}$, basically the cardinality of $\mathcal{S} \rightarrow \mathcal{S}$.

$$2^n = |\mathcal{P}(\mathcal{S})| = 2^{|\mathcal{S}|} = |\{F : \mathcal{S} \rightarrow \{0,1\}\}|$$

characteristic functions of the subsets of $\mathcal{S}$

$$X_A : \mathcal{S} \rightarrow \{0,1\} \quad \begin{cases} \text{if } X_A \in A \text{ then } X_A(x) = 1 \\ \text{if } X_A \notin A \text{ then } X_A(x) = 1 \end{cases}$$

"If a natural number is divisible by 4 then it is divisible by 2"

$$n = 4k \implies n = (2 \cdot 2)k \implies 2 \cdot (2k)$$

The only assumption is that $n$ is divisible by 4 for all natural numbers.

"If 4 is divisible by 4, then 4 is divisible by 2".
This is a type of $true \to true = true$.

But this must be $true$ for all natural numbers: "If 3 is divisible by 4, then 3 is divisible by 2" This is a $false \to false = true$, so $false$ implies $false$ must be $true$.

The last case of implication can be covered by: "If 6 is divisible by 4 then 6 is divisible by 2".
Which is the case $false \to true = true$.

$\square$

We want to prove how we can **encode** the **infinite information** given by a general truth value assignment on all letters of $\mathcal{L}$ in **finite time**.

**Lemma:** let $F \in \mathcal{F}_{\mathcal{L}}$ be a formula and let $v, v' : \mathcal{L} \to \{0, 1\}$ be two (possibly) distinct truth value assignments. Then: if $v(p) = v'(p)$ for all $p \in \mathcal{L}$ letters **actually occurring** in $F$ (occur as a sub-formula, so occur in every $\mathcal{L}$-construction of $F$) then $v(F) = v'(F)$. The only relevant part of an assignment is the **letters actually occurring**, we don't care about all the infinite non-occurring letters.

*Proof.* Structural induction of F

**Base:** If $F = p$, $p \in \mathcal{L}$ then $p$ occurs in $F$, trivially then by hypothesis $v(p) = v'(p)$, then by definition of $\tilde{v}$, $\tilde{v}(p) = \tilde{v}'(p)$.

**Steps:** If $F = (\neg A)$ for some $A \in \mathcal{F_L}$, then by induction hypothesis (I.H.) $\tilde{v}(A) = \tilde{v}'(A)$. Since each $p \in \mathcal{L}$ occurring in $F$ occurs in $A$ too, by definition of $I_\neg : \tilde{v}(\neg A) = \tilde{v}(\neg A)$ then $v(F) = v'(F)$.

If $F = (A \wedge B)$ for some $A, B \in \mathcal{F_L}$ then each $p \in \mathcal{L}$ occurring in $A$ occurs in $F$ too, each $p \in \mathcal{L}$ occurring in $B$ occurs in $F$ too.

Each $p \in \mathcal{L}$ occurring in $F$ occurs in $A$ or in $B$.
So we can apply the I.H. $\tilde{v}(A) = \tilde{v}'(A)$ and $\tilde{v}(B) = \tilde{v}'(B)$. By definition of $I_\wedge$

$$\tilde{v}(A \wedge B) = I_\wedge(\tilde{v}(A), \tilde{v}(B)) = I_\wedge(\tilde{v}'(A), \tilde{v}'(B)) = \tilde{v}'(A \wedge B)$$

$\square$

**Remarks on the lemma:** the lemma allow us **evaluate the behavior of any** $F \in \mathcal{F_L}$ under all assignments (of which there are infinitely many) **just by computing the truth table of** $F$ (finite object, albeit it can be large), whose columns are indexed by the prepositional letters actually occurring in $F$.

If in $F$ are present exactly the propositional letters $p_1, p_2, \dots, p_n$, how many rows does the table for $F$ have? There are $2^n$ rows, it's exponential in the length of $F$. Each row is a possible assignment (function) of $p_1, \dots, p_n \to \{0, 1\}$.

Let $A, B \in \mathcal{F_L}$:

1. $A \to \neg A$: satisfiable, but not a tautology
2. $A \wedge \neg A$: unsatisfiable
3. $A \to (A \vee B)$: tautology

You can check with truth tables (under each column the result of each operand):

| $A$ | $A$ | $\to$ | $\neg$ | $A$ |
|---|---|---|---|---|
| 0 | 0 | 1 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 | 1 | 0 |

| $A$ | $A$ | $\wedge$ | $\neg$ | $A$ |
|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 0 | 0 |
| 1 | 1 | 0 | 0 | 1 | 0 |

| $A$ | $B$ | $A$ | $\to$ | $(A$ | $\vee$ | $B)$ |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 | 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | 1 | 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

Exercise: verify that the following are tautologies

- $A \to (B \to A)$ (called "weakening")
- $(\neg \to A) \to A$ (called "consequentia mirabilis")
- $(A \to B) \vee (B \to A)$ (called "prelinearity")
- $(A \wedge B) \to A$
- $A \to (A \vee B)$
- $A \vee \neg A$ (called "tertium non datur" or "excluded middle")
- $\neg\neg A \to A$ and $A \to \neg\neg A$ (called "double negation laws" or "involutiveness of negation")

Let $A \in \mathcal{F}_{\mathcal{L}}$, $A \to \neg A$, the truth table:

| $A$ | $A$ | $\to$ | $\neg$ | $A$ | |
|---|---|---|---|---|---|
| 0 | 0 | 1 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 | 1 | 0 |

This **does not take into consideration differences between $A$ and simple letters**, but if we instantiate $A$ with a specific formula, for example $A : p \wedge \neg p$ (a contradiction), we can verify that $(p \wedge \neg p) \to \neg(p \wedge \neg p)$ is a tautology.

But if $A : p \vee \neg p$ then

$$(p \vee \neg p) \to \neg(p \vee \neg p)$$

is a contradiction.

A tautology/contradiction remains a tautology/contradiction if you replace letters with arbitrary formulas (the contrary doesn't necessarily hold).

Prelinearity: $(A \to B) \vee (B \to A)$ tautology with $(A, B \in \mathcal{F}_{\mathcal{L}})$. This kinda means that "in every possible world $A \to B$ is true or $B \to A$ is true". Looking at the individual truth tables we can see that it's a tautology.

Excluded middle: $A \vee \neg A$, the refusation of this tautology is the start of other types of logic, not propositional.
This means "in every possible world A is true or A is false", which is clearly a tautology.

## 2  Semantics notions for sets of formulas

Let $\Gamma \subseteq \mathcal{F}_\mathcal{L}$ ($\Gamma$ is a subset of the set of all formulas), then $\Gamma$ is called a **theory**.

**Definition:**  a **theory** $\Gamma \subseteq FL$ is **satisfiable** iff it exists a truth value assignment $v : \mathcal{L} \to \{0,1\}$ such that for all $\gamma \in \Gamma$ then $\tilde{v}(\gamma) = 1$ (in symbols $v \models \gamma$ we mean $v$ satisfies $\gamma$ as a formula) .

**Definition:**  $\Gamma \subseteq \mathcal{F}_\mathcal{L}$ is **unsatisfiable** iff for all assignments $v : \mathcal{L} \to \{0,1\}$ there exists at least $\gamma \in \Gamma$ such that $\tilde{v}(\gamma) = 0$.

$v \not\models \Gamma$ for all $v : \mathcal{L} \to \{0,1\}$. $v \not\models \gamma$ if $\tilde{v}(\gamma) = 0$.

## 2.1  Logical consequence

Of a formula $F \in \mathcal{F}_{\mathcal{L}}$ from a theory $\Gamma \subseteq \mathcal{F}_{\mathcal{L}}$).

**Definition:**  let $\Gamma \subseteq \mathcal{F}_{\mathcal{L}}$, $F \in \mathcal{F}_{\mathcal{L}}$ , then $F$ **is a logical consequence of** $\Gamma$, $\Gamma \models F$, iff for every assignment $v : \mathcal{L} \to \{0, 1\}$ satisfying $\Gamma$ it holds that $v$ satisfies $F$ too:

$$v \models \Gamma \implies v \models F$$

To express the fact that a given formula $F \in \mathcal{F}_{\mathcal{L}}$ **is a tautology** by means of the notion of logical consequence there are several ways, for example

$$T \models F$$

with $T$ being an always true constant.
Another solution: let $\Omega$ be a set of tautologies:

$$\Omega \models F$$

Or even:

$$\emptyset \models F$$

since you can't falsify a formula in the $\emptyset$, there's no formula.
For a **tautology** is also commonly used:

$$\models F$$

with no operand on the left side.

### 2.1.1 Fundamental semantic properties of theories with respect to logical consequence

**Lemma:** let $\Gamma \subseteq \mathcal{F}_{\mathcal{L}}$ be a theory and $A \in \mathcal{F}_{\mathcal{L}}$, then $\Gamma \models A$ **iff** $\Gamma \cup \{\neg A\}$ is unsatisfiable.

*Proof.* $\Gamma \models A$ **iff** (by def of $\models$) for all assignments $v : \mathcal{L} \to \{0,1\}$, $\tilde{v}(\gamma) = 1$ for all $\gamma \in \Gamma$ implies $\tilde{v}(A) = 1$.

**Iff** for all $v : \mathcal{L} \to \{0,1\}$ either it is not the case that $\tilde{v}(\gamma) = 1$ for all $\gamma \in \Gamma$, or $\tilde{v}(A) = 1$.

**Iff** for all $v : \mathcal{L} \to \{0,1\}$ there exists $\gamma \in \Gamma$ such that $\tilde{v}(\gamma) = 0$ or $\tilde{v}(\neg A) = 0$.

**Iff** (by definition of unsatisfiability) $\Gamma \cup \{\neg A\}$ is unsatisfiable.

$\square$

Basically, if $A$ is a logical consequence of $\Gamma$ then there's no case in which $\Gamma \cup \{\neg A\}$ can be true, since for all assignments in which $\Gamma$ is satisfied $A$ is also satisfied and thus $\neg A$ is always false.

**Lemma (Semantical deduction Lemma):** Let $P, Q \in \mathcal{F}_\mathcal{L}$, $\{P\} \models Q$ iff $P \to Q$. The logical consequence relation can be internalized using the logical implication connective.

*Proof.* $P \models Q$ **iff** (by definition of $\models$) for all $v : \mathcal{L} \to \{0, 1\}$, $\tilde{v}(P) = 1$ implies $\tilde{v}(Q) = 1$.

**Iff** (by definition of $I_\to$) $\tilde{v}(P \to Q) = 1$ for all $v : \mathcal{L} \to \{0, 1\}$ that is $P \to Q$ is a tautology.

$\square$

Saying $P \models Q$ or $P \to Q$ is kinda the same.

**Theorem (Semantical deduction):** Let $\Gamma \subseteq \mathcal{F}_\mathcal{L}$ and $P, Q \in \mathcal{F}_\mathcal{L}$, then $\Gamma \cup \{P\} \models Q$ (which will be also written as $\Gamma, P \models Q$) iff $\Gamma \models P \to Q$.

*Proof.* $\Gamma, P \models Q$ **iff** (by definition of $\models$) for all assignments $v : \mathcal{L} \to \{0, 1\}$ such that $v \models \Gamma$ and $\tilde{v}(P) = 1$ (all assignments that satisfy both $\Gamma$ and $P$) then $\tilde{v}(Q) = 1$ (also satisfy $Q$).

**Iff** (by definition of $I_\to$) for all $v : \mathcal{L} \to \{0, 1\}$ such that $v \models \Gamma$ it holds that $\tilde{v}(P \to Q) = 1$.

**Iff** (by definition of $\models$) $\Gamma \models P \to Q$.

$\square$

This is a stronger version of the lemma presented earlier.

## 2.2 Compactness Theorem

It's one of the theorems that lies at the foundation of logic.

Let $\Gamma \subseteq \mathcal{F}_{\mathcal{L}}$, $\Gamma$ **is satisfiable iff** for all $\Gamma' \subseteq \Gamma$, with $\Gamma'$ finite, then $\Gamma'$ **is satisfiable**. Every subset of a set of satisfiable formulas is satisfiable.

Do we *really* need to consider the case $\Gamma$ is infinite (contains infinitely many formulas)? Yeah, kinda, it will be shown that in first order logic a single formula may contain the amount of information of infinitely many propositional formulas (whatever this means).

Compactness theorem allows us to **reduce the analysis of satisfiability** of a possibly infinite theory **to the analysis of its finite sub-theories**, but observe that **there are infinitely many finite sub-theories of an infinite theory**.

Rewriting the theorem: $\Gamma \subseteq \mathcal{F}_{\mathcal{L}}$ **is satisfiable iff** for all $\Gamma' \subseteq_\omega \Gamma$ then $\Gamma'$ is satisfiable (with $\subseteq_\omega$ meaning finite subset). The non-trivial case is when $\Gamma$ is infinite.

*Proof.* Proving the statement **from left to right** is trivial, if $v \models \Gamma$ then, clearly, by the definition of satisfiability and the notion of subset, $v \models \Gamma'$, for each $\Gamma' \subseteq \Gamma$ too (not even $\subseteq_\omega$, just any subset, even infinite).

$\square$

Proving right to left is not so trivial.

Consider the statement **equivalent** to the theorem **obtained via contraposition** ($A \to B$ says the same thing as $\neg B \to \neg A$). Again from left to right: if there is $\Gamma' \subseteq_\omega$ is unsatisfiable then clearly $\Gamma$ is unsatisfiable.

From right to left: if $\Gamma$ **is unsatisfiable**, then there **exists** $\Gamma' \subseteq_\omega \Gamma$ such that $\Gamma'$ is **unsatisfiable**.

**Some terminology** before the proof: we say that a given theory $\Gamma \subseteq \mathcal{F}_\mathcal{L}$ is **finitely satisfiable** (fin sat) **iff** for all finite sub-theories $\Gamma' \subseteq_\omega \Gamma$, $\Gamma'$ **is satisfiable**.

The proof of the non trivial part consists of $\Gamma$ fin sat, implies $\Gamma$ sat.

*Proof.* We need some facts before the proof.

**Preparation:** we fix any **arbitrarily chosen listing** $F_1, F_2, ..., F_k, ...$ (enumerably infinite, without repetitions) of all formulas $\in \mathcal{F}_\mathcal{L}$. To concretely implement such an enumeration we have two issues to face:

- $1^{st}$ issue: the **alphabet of** $\mathcal{F}_\mathcal{L}$: $\{\wedge, \vee, \neg, \rightarrow\} \cup \{), (\} \cup \mathcal{L}$. This contains an infinite set an thus is an **infinite alphabet**. To **switch to a finite alphabet**:

$$\{\wedge, \vee, \neg, \rightarrow\} \cup \{), (\} \cup \{p, l\}$$

  with $p_k \in \mathcal{L}$ written as $p_k = plll...l$, with $k$ many $l$s (just a unary encoding, we just need some kind of finite encoding, let it be a unary encoding, binary, base 10, hex, ...).

- $2^{nd}$ issue: take the alphabet $\{a, b, ..., \}$: list all the words on this alphabet, but they are infinite and we must define an order. If we use a lexicographic ordering we get:

| |
|---|
| $\epsilon$ (empty word) |
| $a$ |
| $aa$ |
| $a...aa$ |
| ... |
| $b \leftarrow$ never reached |

So the letter $b$ appears after an infinite amount of words. Using a short-lex ordering might be fairer (each word appears at a finite stage):

| |
|---|
| $\epsilon$ (empty word) |
| $a$ |
| ... |
| $z$ |
| $aa$ |
| $ab$ |
| ... |
| $zz$ |

First word of length 1, then length 2, ecc.

**The construction:** we **define** an **infinite succession of theories** (sets of formulas) as follows: the succession

$$D_0, D_1, \ldots, D_k, \ldots$$

Which is inductively defined as follows:

- **Base:**
$$D_0 := \Gamma$$

- **Step:**

$$D_{n+1} := \begin{cases} D_n \cup \{F_{n+1}\} & \text{if } D_n \cup \{F_{n+1}\} \text{ is fin sat} \\ D_n \cup \{\neg F_{n+1}\} & \text{otherwise} \end{cases}$$

And we also define:
$$D := \bigcup_{i \in \mathbb{N}} D_i$$

Note that this definition is well given and does not directly imply that $D_n \cup \{\neg F_{n+1}\}$ is fin sat, but just that $D_n \cup \{F_{n+1}\}$ is not fin sat.

Now we shall prove some facts about the sequence

$$\Gamma = D_0 \subseteq D_1 \subseteq D_2 \subseteq \ldots \subseteq D_k \subseteq \ldots \subseteq D$$

To prove $\Gamma$ fin sat implies $\Gamma$ sat, we first need to prove some facts about the sequence above.

**Fact 1: every member of the sequence $D_n$** (for all $n \in \mathbb{N}$) is himself **fin sat**.

*Proof.* By induction on $n$

- **Base:** $n = 0$, $D_0$ is fin sat, by definition, since $D_0 = \Gamma$ and $\Gamma$ is fin sat.
- **Step:** assume that Fact 1 is true for $D_0, D_1, \ldots, D_n$ and prove it for $D_{n+1}$. To prove this we shall proceed by contradiction. So we assume $D_{n+1}$ is not fin sat with the aim of finding a contradiction.

Absurdum Hypothesis: $D_{n+1}$ is not fin sat.

Hence by the definition of the sequence, $D_n \cup \{F_{n+1}\}$ and $D_n \cup \{\neg F_{n+1}\}$ are both NOT fin sat.

Now we use the definition of fin sat, they are not fin sat

    **iff** there exists
$$D' \subseteq_\omega D_n \cup \{F_{n+1}\}$$
such that $D'$ is unsatisfiable.

    **Iff** there exists
$$D'' \subseteq_\omega D_n \cup \{\neg F_{n+1}\}$$
such that $D''$ is unsatisfiable.

"Lazy" version, we can assume, without loss of generality that $F_{n+1} \in D'$, $\neg F_{n+1} \in D''$. We can argue that they are already in there, otherwise we can add them without major problems (if it's unsatisfiable adding something keeps it that way).

The actual version: if $F_{n+1} \notin D'$ then $D' \subseteq D_n$, but $D_n$ is fin sat, so $D'$ is fin sat, which is a contradiction.

That is we can display $D'$ and $D''$ as follows
$$D' = E' \cup \{F_{n+1}\}, \quad D'' = E'' \cup \{\neg F_{n+1}\}$$
with $E' = D' \setminus \{F_{n+1}\}$ and $E'' = D'' \setminus \{\neg F_{n+1}\}$.

We observe that by construction that $E', E'' \subseteq_\omega D_n$ they are subsets of $D_n$.

**Note**: if we take the **theory** $E' \cup E'' \cup \{F_{n+1}\}$ this is **unsatisfiable**, since it contains $D'$. For the same reason $E' \cup E'' \cup \{F_{n+1}\}$ is **also unsatisfiable** since it contains $D''$.

But $E' \cup E''$ is a finite subset of $D_n$, $\subseteq_\omega D_n$, so they **must be fin sat** since we know that $D_n$ is fin sat by I.H. $E' \cup E''$ **is satisfiable**.

This **satisfiable** set **differs** from the **unsatisfiable** ones by **one formula**, notably the same formula.

If we find an assignment that satisfies $F_{n+1}$ then $D'$ is satisfiable, otherwise $\neg F_{n+1}$ is satisfied and $D''$ becomes satisfiable.

So, we have:
- $E' \cup E''$ **satisfiable**
- $E' \cup E'' \cup \{F_{n+1}\}$ **unsatisfiable** (†)
- $E' \cup E'' \cup \{\neg F_{n+1}\}$ **unsatisfiable** (††)

And there exists (by definition of satisfiability) $v : \mathcal{L} \to \{0, 1\}$ such that $v \models E' \cup E''$ ($\tilde{v}(F) = 1$ for all $F \in E' \cup E''$). We can have a **truth assignment** that **satisfies** all formulas in that **set**.

What about $\tilde{v}(F_{n+1})$?
- if it's 0 it will satisfy $\tilde{v}(\neg F_{n+1})$ and then $v \models E' \cup E'' \cup \{\neg F_{n+1}\}$, **contradicting** (††)
- If it's 1 it will satisfy $\tilde{v}(F_{n+1})$ and then $v \models E' \cup E'' \cup \{F_{n+1}\}$, **contradicting** (†)

And, by **bivalence**, there are **no other cases to consider**. We have reached in any case the contradiction that proves the induction step ($D_{n+1}$ is actually fin sat).

This concludes the proof by induction of Fact 1.

We have proved that **each set** in our construction $D_i$ **is fin sat**.

$\square$

**Fact 2:** remembering the definition of $D$, $D$ is fin sat. We proved that **each set is fin sat**, now we need to **prove** that **their union is fin sat**

$$D = \bigcup_{i \in \mathbb{N}} D_i \quad \text{id fin sat}$$

Do we *really* need to prove this? Yes, there are some properties of some sets, such as finiteness, that don't live into their union (with a collection of finite sets, their union might not be a finite set). **Properties of the members don't automatically carry over to their union**.

*Proof.* Let $D'$ be a finite subset of $D$ ($D' \subseteq_\omega D$). It is sufficient to prove that $D'$ is satisfiable.

We can display the finite set $D'$ by listing its formulas:

$$D' = \{F_{i1}, F_{i2}, ..., F_{im}\} \text{ for some } m \in \mathbb{N}$$

We give these indexes since we don't know exactly what formulas are inside $D'$, but there must be a maximum, whatever that number is and whatever the numbers in the set are.

Let $k$ be the maximum index

$$k = \max_{j=1}^{m}(ij)$$

Then

$$D' \subseteq \{F_1, F_2, ..., F_k\} \cap D \subseteq D_k$$

The set $D'$ must be a subset of all the formulas between 1 and its highest index $k$, which is a subset of $D_k$, by definition f $D_k$.

So $D' \subseteq_\omega D_k$ (as $D_k$ is defined in our sequence), but by Fact 1 $D_k$ is fin sat. So $D'$ is satisfiable since it's one of its sub-theories.

$\square$

**Fact 3:** For each formula $F_t \in \mathcal{F}_\mathcal{L}$, **exactly one among** $F_t$ and $\neg F_t$ **belongs to** $D$.

*Proof.* By construction of the sequence, $F_t \in D_t$ or $\neg F_t \in D_t$. Since $D_t \subseteq D$: $F_t \in D$ or $\neg F_t \in D$. This proves that at least one is in the set.

We now have to exclude that both of them are in the set: ($\{F_t, \neg F_t\} \subseteq D$). But $D$ is satisfiable, and clearly $\{F_t, \neg F_t\}$ would contradict Fact 2 since it can't be satisfiable, by the definition of $I_\neg$.

So $F_t \in D$ or $\neg F_t \in D$ and $\{F_t, \neg F_t\} \notin D$. Exactly one is $\in D$.

$\square$

**Construction:** we define the following **assignment** $v_D : \mathcal{L} \to \{0, 1\}$. For all $p \in \mathcal{L}$, $v_D(p) = 1$ iff $p \in D$.

Notice that $v_D$ is **well defined** (either $p \in D$ or $p \notin D$ for each possible $p \in \mathcal{L}$).

**Fact 4:** $v_D \models D$ (for all $F \in D$, $\tilde{v}_D(F) = 1$); it's an assignment that satisfies $D$. If we can **prove this** then we have **proved that $v_D \models \Gamma$ since** $\Gamma \subseteq D$ (by construction of $D$). So $\Gamma$ is sat.

*Proof.* By structural induction, we prove that for all formulas $F \in \mathcal{F}_\mathcal{L}$: $\tilde{v}_D(F) = 1$ **iff** $F \in D$.

Notice: it would suffice to prove just one side of the iff, namely the side that says if $F \in D$ then $\tilde{v}_D(F) = 1$ (but the stronger proof is easier). We ask for the biconditional (iff) for better using the I.H.

**Base:** $F = p$, with $p \in \mathcal{L}$: so the statement we need to prove is the very definition of $\tilde{v}_D(p)$:

$$\tilde{v}_D(p) = v_D(p) = 1 \ \text{ iff } \ p \in D$$

**Step(s):** we need to consider each way that we can decompose a formula and prove that it's $\in D$:

- $F = (\neg G)$, $F$ is the negation of some formula $G$.
  $\tilde{v}_D(F) = 1$ **iff** (by definition of $I_\neg$) $\tilde{v}_D(G) = 0$ **iff** (by I.H.) $G \notin D$ **iff** (by Fact 3) $\neg G \in D$ **iff** $F \in D$ (since $\neg G$ is $F$).

- $F = (G \wedge H)$.
  $\tilde{v}_D(F) = 1$ **iff** $\tilde{v}_D(G \wedge H) = 1$ **iff** (by definition of $I_\wedge$) $\tilde{v}_D(G) = 1$ and $\tilde{v}_D(H) = 1$ **iff** (by I.H.) $G \in D$ and $H \in D$ **iff** $G \wedge H \in D$ **iff** $F \in D$.
  To say that $G \wedge H \in D$ we can observe (by contradiction) that if $G \wedge H \notin D$ **iff** (by Fact 3) $\neg(G \wedge H) \in D$, then $\{G, H, \neg(G \wedge H)\} \subseteq_\omega D$ and this is a contradiction since by Fact 2 $D$ is fin sat, and this subset is unsatisfiable.
  Then if $G \wedge H \in D$ then $G, H \in D$ (otherwise $G \notin D$ or $H \notin D$). Assume $G \notin D$ (the case $H \notin D$ is analogous), by Fact 3, $\neg G \in D$ and then $\{G \wedge H, \neg G\} \subseteq_\omega D$, which is unsatisfiable, contradicting Fact 2 again. So $F \in D$.

- $F = (G \vee H)$.
  $\tilde{v}_D(G \vee H) = 1$ iff (by definition of $I_\vee$) $\tilde{v}_D(G) = 1$ or $\tilde{v}_D(H) = 1$ **iff** (by I.H.) $G \in D$ or $H \in D$ **iff** $G \vee H \in D$ **iff** $F \in D$.
  If $G \in D$ or $H \in D$ then $(G \vee H) \in D$. By contradiction $(G \vee H) \notin D$, then, by Fact 3, $\neg(G \vee H) \in D$, then
    - if $G \in D$ then $\{G, \neg(G \vee H)\} \subseteq_\omega D$, which contradicts Fact 2
    - if $H \in D$ then $\{H, \neg(G \vee H)\} \subseteq_\omega D$, which contradicts Fact 2
  If $(G \vee H) \in D$ then $G \in D$ or $H \in D$. By contradiction $G \notin D$ and $H \notin D$ and thus $\{\neg G, \neg H, (G \vee H)\} \subseteq_\omega D$, which is unsatisfiable and contradicts Fact 2.

- Prove implication for fun (remember that $G \to H$ has the same truth table as $(\neg G) \vee H$)

$\square$

We have proved Fact 4: $v_D \models D$.

Then a fortiori (for stronger reasons) $v_D \models \Gamma$ (the assignment satisfies $D$, which is larger that $\Gamma$, so $v_D$ must satisfy $\Gamma$).

So we have proved that:

$$\Gamma \ \text{fin sat} \ \Leftrightarrow \ \Gamma \ \text{sat}$$

$\square$

By **contraposition**:

$$\Gamma \ \text{unsat} \ \implies \ \text{there exists} \ \Gamma' \subseteq_\omega \Gamma, \ \Gamma' \ \text{unsat}$$

**Remarks on Compactness Theorem:**

$$\Gamma \subseteq \mathcal{F}_\mathcal{L} \quad \Gamma' \subseteq \Gamma$$

We **proved the theorem** not by looking inside $\Gamma$, but by **enlarging it** with our sequence of $D_i$s, and **proved that** $v_D \models D$, and so $v_D \models \Gamma$, since $\Gamma \subseteq D$.

$D$ is "large" and satisfiable. We **cannot expand $D$ further without losing** the property of $D$ **being satisfiable**.

*Proof.* First of all, is $D$ everything? Does it contain it all? No, it doesn't, look at Fact 3, so $D \neq \mathcal{F}_\mathcal{L}$.

Let $F_t \notin D$. By Fact 3: $\overline{D} = D \cup \{F_t\}$, and thus $\neg F_t \in D$.
Then $\{F_t, \neg F_t\} \subseteq_\omega \overline{D}$ so $\overline{D}$ is unsatisfiable.
If I add anything not already in $D$, the negation of that formula must already be in $D$ so the set cannot remain satisfiable if I add anything else.

$\square$

$D$ is a **maximal expansion**, with respect to **satisfiability**, of $\Gamma$.

We call a set $E$ of formulas **maximally satisfiable** if it has the same property of $D$: **iff** $E$ is a set and for all $F \in \mathcal{F}_\mathcal{L}$, $F \notin E$ the set $E \cup \{F\}$ is unsatisfiable (if I add anything the set cannot remain satisfiable).

$D$ is **a** maximal expansion of $\Gamma$, there can be **many**, in our proof $D$ is fixed, after fixing the listing of formulas.

Let us see that there are **different maximal expansions** of $\Gamma$. For instance $\Gamma = \{p_{27}\}$, we have 2 listings:

- listing 1: $p_{13}$, $p_{14}$
- listing 2: $\neg p_{13}$, $\neg p_{14}$

Then for listing 1: $D_1$ is expanded with $p_{13}$ while the $D_1$ made from listing 2 is expanded with $\neg p_{13}$ so, in the final set $D$, for listing 1 $p_{13} \in D$ while for listing 2 $\neg p_{13} \in D$. We can expand our set in different ways, based on which listing I choose.

If we reason about the **information contained** inside a **maximally satisfiable theory** $D$ we notice that it carries the **same amount of information** as a **truth value assignment** $v : \mathcal{L} \to \{0, 1\}$.

$D$ max sat theory, from this we defined $v_D$ as $v_D(p) = 1$ iff $p \in D$ for all $p \in \mathcal{L}$. We can go the other way around, **starting from any assignment** $v : \mathcal{L} \to \{0, 1\}$ and **construct the set of formulas $D_v$**:

$$D_v := \{F \in \mathcal{F}_\mathcal{L} : \tilde{v}(F) = 1\}$$

This way

- $D_v$ is a max sat theory
- $v_{D_w} = w$, with $w$ truth assignment $w :\to \{0, 1\}$
- same thing with: $D_{v_E} = E$ with $E$ a max sat theory

Associated with every max sat theory there is an assignment satisfying the theory ($v_D \models D$), and similarly for every assignment there can be a max sat theory satisfied by that assignment.

How can we tell if:

$$(\Gamma \subseteq \mathcal{F}_{\mathcal{L}}, \; A \in \mathcal{F}_{\mathcal{L}}) \quad \Gamma \models A?$$

All formulas in $\Gamma$ must satisfy also $A$, so $A$ it's a logical consequence.

Already proven Lemma: $\Gamma \models A$ iff we can **reduce this problem to** the unsatisfiability problem: $\Gamma \cup \{\neg A\}$ **is unsatisfiable**.

Assume $\Gamma$ is a finite theory ($\Gamma$ contains finitely many formulas)

$$\Gamma = \{B_1, B_2, ..., B_u\}$$

There are finitely many formulas, so there's a last element $B_u$

$$\Gamma \models A \text{ iff}$$
$$\{B_1, B_2, ..., B_u\} \models A \text{ iff}$$
$$\{B_1, B_2, ..., B_u, \neg A\} \text{ is unsat, as a theory, iff}$$

by the definition of $\wedge$ $(I_\wedge)$

$$B_1 \wedge B_2 \wedge ... \wedge B_u \wedge \neg A \text{ is unsat, as a formula}$$

And this is decidable in finite time, however long it may be.

What if $\Gamma$ is infinite? The conjunction of infinitely many things it's not a formula and its satisfiability it's not decidable in finite time. So

$$\Gamma \cup \{\neg A\} \text{ is unsat}$$

By compactness theorem, there is

$$\Gamma' \subseteq_\omega \Gamma \cup \{\neg A\} \text{ is unsat}$$

Good news: we can have a finite theory which is unsatisfiable. Bad news: we don't know how to find it since there are infinitely many.
$\rightarrow$ "Semidecidability". If the input is infinite, it's improper speaking of decidability, it's semidecidability. We may be able to tell if it's satisfiable in finite time, or the algorithm could not end.

## 2.3 Semantic Equivalence (logical equivalence)

Are these the same?
$$A \wedge B \qquad B \wedge A$$

These are syntactically different, they are different strings of symbols.

Semantically, **for all assignments** $v : \mathcal{L} \to \{0, 1\}$ it holds that $\tilde{v}(A \wedge B) = \tilde{v}(B \wedge A)$, by the definition of $I_\wedge$. They always **get the same resulting truth values**.

### 2.3.1 Equivalence relation

**What is a relation?** An $n$-ary relation $R$ on a set $S$ is just a subset of the $n$-th Cartesian power $S^n$ of $S$:

$$R \subseteq S^n$$

A relation, mathematically, is just this.

The **equivalence relation**: considering $R$ on a set $S$, $R$ is a **binary relation** $(R \subseteq S^2)$ such that it **satisfies 3 conditions**:

- $R$ is **reflexive**: for all $s \in S$, $(s, s) \in R$
- $R$ is **symmetric**: for all $s_1, s_2 \in S$ if $(s_1, s_2) \in R$ then also $(s_2, s_1) \in R$
- $R$ is **transitive**: for all $s_1, s_2, s_3 \in S$ if $(s_1, s_2) \in R$ and $(s_2, s_3) \in R$ then $(s_1, s_3 \in R)$

**Equivalence classes:** Let $R \subseteq S^2$ be an equivalence relation, then for each $s \in S$ the **equivalence class** of $s$, called $([s]_R$ or $s/R)$ is **defined as**

$$[s]_R := \{t \in S : (s, t) \in R\}$$

Basically, all elements equivalent to $s$.

Observe: if $(s, t) \in R$ then $[s]_R = [t]_R$ and vice versa, if $[s]_R = [t]_R$ then $(s, t) \in R$.

**Partitions of a set:** A partition of $S$ is a set $\{B_1, B_2, \ldots, B_k, \ldots\} = \{B_i : i \in I\}$ of subsets of $S$ (not necessarily finite, $B_i \subseteq S$) such that:

- $B_i \cap B_j = \emptyset$ for all $i \neq j \in I$
- $\bigcup_{i \in I} B_i = S$

A partition is a way of **cutting $S$ into blocks**.

Every **equivalence relation $R \subseteq S^2$ determines uniquely a partition $P_R$ of $S$**

$$P_R = \{[s]_R : s \in S\}$$

The blocks of $P_R$ are the equivalence classes of $R$.

We can go the other way around, every **partition $P$ of $S$ determines uniquely an equivalence relation $R_P$ on $S$ ($R_P \subseteq S^2$)**

$$R_p : (s, t) \in R_P \text{ iff } s, t \text{ belong to the unique same block } B_i \in P$$

### 2.3.2 Definition of Semantic equivalence

Let us verify the the following **is an equivalence relation on $\mathcal{F}_\mathcal{L}$**

$$\equiv \subseteq \mathcal{F}_\mathcal{L}^2$$

For each $(A, B) \in \mathcal{F}_\mathcal{L}^2$ (or simpler, $A, B \in \mathcal{F}_\mathcal{L}$), then $(A, B) \in \equiv$, also written as $A \equiv B$ ($A$ is **logically equivalent to** $B$) **iff**, for any $v : \mathcal{L} \to \{0, 1\}$, $\tilde{v}(A) = \tilde{v}(B)$.

For all $A \in \mathcal{F}_\mathcal{L}$:

- $\equiv$ is **reflexive**: for all $A \in \mathcal{F}_\mathcal{L}$, $A \equiv A$, for each $v : \mathcal{L} \to \{0, 1\}$ $\tilde{v}(A) = \tilde{v}(A)$
- $\equiv$ is **symmetric**: for all $A, B \in \mathcal{F}_\mathcal{L}$, $A \equiv B$ implies $B \equiv A$, for all $v : \mathcal{L} \to \{0, 1\}$: $\tilde{v}(A) = \tilde{v}(B)$ implies $\tilde{v}(A) = \tilde{v}(B)$
- $\equiv$ is **transitive**: for all $A, B, C \in \mathcal{F}_\mathcal{L}$, $A \equiv B$ and $B \equiv C$ implies $A \equiv C$, for all $v : \mathcal{L} \to \{0, 1\}$: $\tilde{v}(A) = \tilde{v}(B)$ and $\tilde{v}(B) = \tilde{v}(C)$ implies $\tilde{v}(A) = \tilde{v}(C)$

$\equiv$ **determines a partition of** $\mathcal{F}_\mathcal{L}$. We subdivided the world in such a way that formulas that say the same thing are in the same block

$$P_\equiv = \{[A]_\equiv : A \in \mathcal{F}_\mathcal{L}\}$$

$$\mathcal{F}_\mathcal{L}/_\equiv = \{[A]_\equiv : A \in \mathcal{F}_\mathcal{L}\}$$

$[A \wedge B]_\equiv = [B \wedge A]_\equiv$ says that the formulas are equivalent, which is the same as saying $A \wedge B \equiv B \wedge A$.

Two formulas are **logically equivalent** if they have the **same truth value under any assignment**.

In general if $R \subseteq S^2$ is an equivalence relation, we write $S/_R = \{[S]_R : s \in S\}$. This is called "$S$ modulo $R$" or "the quotient of $S$ over $R$".

We eventually will enrich the set of logical equivalence classes $\mathcal{F}_\mathcal{L}/_\equiv$ with operations which shall confer to the enriched set the status of an algebraic structure (namely: Boolean algebra).

## 2.4   Substitution (letter with a formula)

**Substitution** of a **propositional letter with a formula** (inside a formula).   We want to get new formula from another one, while remaining semantically equivalent.

For each $A \in \mathcal{F}_{\mathcal{L}}$, for each $p \in \mathcal{L}$, for each $B \in \mathcal{F}_{\mathcal{L}}$, let us define the formula obtained by **replacing in $A$ each occurrence of $p$ with $B$**, $A\,[B/p] \in \mathcal{F}_{\mathcal{L}}$. Define it inductively as follows:

- **Base:** $A = q$, $Q \in \mathcal{L}$, define

$$q[B/p] := \begin{cases} B & q = p \\ q & q \neq p \end{cases}$$

  If $q = p$ then it becomes $B$, otherwise nothing changes.
- **Step:** cases, if we have to substitute:

$$(\neg A)[B/p] := \neg(A[B/p])$$

$$(A_1 \wedge A_2)[B/p] := A_1[B/p] \wedge A_2[B/p]$$

$$(A_1 \vee A_2)[B/p] := A_1[B/p] \vee A_2[B/p]$$

$$(A_1 \rightarrow A_2)[B/p] := A_1[B/p] \rightarrow A_2[B/p]$$

**Lemma (substitution):** let $v : \mathcal{L} \to \{0,1\}$ be an assignment and $S, T \in \mathcal{F}_\mathcal{L}$. If $\tilde{v}(S) = \tilde{v}(T)$, then for all $A \in \mathcal{F}_\mathcal{L}$, and all $p \in \mathcal{L}$

$$\tilde{v}(A[{}^S\!/_p]) = \tilde{v}(A[{}^T\!/_p])$$

*Proof.* By induction on $A$:

- **Base:** $A = q$, $q \in \mathcal{L}$, two cases:
  - $q \neq p$, then $q[{}^S\!/_p] = q = q[{}^T\!/_p]$, then

  $$\tilde{v}(q[{}^S\!/_p]) = \tilde{v}(q) = \tilde{v}(q[{}^T\!/_p])$$

  - $q = p$, then $q[{}^S\!/_p] = S$ and $q[{}^T\!/_p] = T$, so

  $$\tilde{v}(q[{}^S\!/_p]) = \tilde{v}(S) = \tilde{v}(T) = \tilde{v}(q[{}^T\!/_p])$$

- **Step:** cases are:
  - $A = \neg B$, by I.H. $\tilde{v}(B[{}^S\!/_p]) = \tilde{v}(B[{}^T\!/_p])$ and then

  $$\tilde{v}((\neg B)[{}^S\!/_p]) = I_\neg(\tilde{v}(B[{}^S\!/_p])) = I_\neg(\tilde{v}(B[{}^T\!/_p])) = \tilde{v}((\neg B)[{}^T\!/_p])$$

  - $A = (B \wedge C)$
  - $A = (B \vee C)$
  - $A = (B \to C)$

  The other cases are left as an exercise for the reader.

  $\square$

**Theorem (substitution):** Let $S, T \in \mathcal{F}_\mathcal{L}$, if $S \equiv T$, then for all $A \in \mathcal{F}_\mathcal{L}$ and all $p \in \mathcal{L}$:

$$A[{}^S\!/_p] \equiv A[{}^T\!/_p]$$

If in a single formula we **replace a formula with an equivalent** one we get a **distinct** formula **but still equivalent**.

*Proof.* For all $v : \mathcal{L} \to \{0,1\}$ it holds that $\tilde{v}(s) = \tilde{v}(T)$, since $S \equiv T$. So by the substitution Lemma, for all $v : \mathcal{L} \to \{0,1\}$ we have $\tilde{v}(A[{}^S\!/_p]) = \tilde{v}(A[{}^T\!/_p])$, and then, by definition of $\equiv$

$$A[{}^S\!/_p] \equiv A[{}^T\!/_p]$$

$\square$

**Theorem:** Given $F, G, H \in \mathcal{F}_\mathcal{L}$ the following are logical equivalences:

- **Idempotence**: $F \wedge F \equiv F$, $F \vee F \equiv F$
- **Commutativity**: $F \vee G \equiv G \vee F$, $F \wedge G \equiv G \wedge F$
- **Associativity**: $F \wedge (G \wedge H) \equiv (F \wedge G) \wedge H$, $F \vee (G \vee H) \equiv (F \vee G) \vee H$
- **Distributivity**: $F \wedge (G \vee H) \equiv (F \wedge G) \vee (F \wedge H)$, $F \vee (G \wedge H) \equiv (F \vee G) \wedge (F \vee H)$
- **Absorption**: $F \vee (F \wedge G) \equiv F$, $F \wedge (F \vee G) \equiv F$

Exercise: prove that idempotence $F \vee F \equiv F$, $F \wedge F \equiv F$ follows only by absorption (using substitution theorem).

- **Double negation** (involutiveness of negation): $\neg\neg F \equiv F$
- **De Morgan**:
$$\begin{cases} \neg(F \vee G) \equiv \neg F \wedge \neg G \\ \neg(F \wedge G) \equiv \neg F \vee \neg G \end{cases}$$

- **Interdefinability**:
$$\begin{cases} F \to G \equiv (\neg F) \vee G \\ F \to G \equiv \neg(F \wedge \neg G) \\ F \vee G \equiv (\neg F) \to G \\ F \wedge G \to \neg(F \to \neg G) \\ F \vee G \equiv \neg(\neg F \wedge \neg G) \\ F \wedge G \equiv \neg(\neg F \vee \neg G) \end{cases}$$

*Proof.* Just one case (the remaining ones are left as an exercise for the reader). We want to show that

$$\neg(F \vee G) \equiv \neg F \wedge \neg G$$

For all $v : \mathcal{L} \to \{0, 1\}$ $\tilde{v}(\neg(F \vee G)) = 1$

    **iff** by the semantics of $\neg$: $I_\neg$: $\tilde{v}(F \vee G) = 0$
    **iff** by the semantics of $\wedge$: $\tilde{v}(F) = 0$ and $\tilde{v}(G) = 0$
    **iff** by the semantics of $\neg$: $\tilde{v}(\neg F) = 1$ and $\tilde{v}(\neg G) = 1$
    **iff** by the semantics of $\wedge$: $\tilde{v}(\neg F \wedge \neg G) = 1$

By bivalence $\neg(F \vee G) \equiv \neg F \vee \neg G$ (we've proved that it holds for 1, the only other possible value is 0 so we don't need to prove that).

$\square$

## 2.5 Derived connectives

We define these operators as **shortened versions of longer formulas**.

We define $F \leftrightarrow G$, as $(F \rightarrow G) \wedge (G \rightarrow F)$. Specify

$$I_\leftrightarrow : \{0,1\}^2 \rightarrow \{0,1\}$$

| $a$ | $b$ | $a \leftrightarrow b$ |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

**Observation:** $\models F \leftrightarrow G$ (is a tautology) **iff** $F \equiv G$ **iff** $\models \neg F \leftrightarrow \neg G$ **iff** $\models G \leftrightarrow F$ **iff** $\models \neg G \leftrightarrow \neg F$.

**Observation:** $F \models G$ and $G \models F$ **iff** $F \equiv G$.

We define other derived connectives $\bot, \top$. Technically they are zero-ary connectives (and as such they are formulas in themselves), they're **logical constants**, don't need arguments to be complete to a formula.

$$\bot, \top \in \mathcal{F}_\mathcal{L}$$
$$I_\bot : \{0,1\}^0 \rightarrow \{0,1\} \quad \text{identify with } 0$$
$$I_\top : \{0,1\}^0 \rightarrow \{0,1\} \quad \text{identify with } 1$$

The $\{0,1\}^0$ represents the singleton set, so we can **identify the function with the result**. So
$$I_\bot := 0 \quad I_\top := 1$$

The $\top$ is in the **equivalence class of tautologies**:

$$\top \equiv A \quad \text{iff } A \text{ is any tautology } (\models A), \text{ e.g. } \top := p \rightarrow p$$

And $\bot$ is in the **equivalence class of contradictions**:

$$\bot \equiv A \quad \text{iff } A \text{ is any contradiction } (\models \neg A), \text{ e.g. } \bot := p \wedge \neg p$$

Some things **about these constants**:

$$\neg\top \equiv \bot, \quad \neg\bot \equiv \top$$

If $\models A$

$$\neg A \equiv A \to \bot$$
$$A \equiv \top \to A$$

Some other equivalences:

- **Complementation**

$$A \wedge \neg A \equiv \bot$$
$$A \wedge \neg A \equiv \top$$

- **Unit** and **neutral elements**

$$A \wedge \bot \equiv \bot$$
$$A \vee \bot \equiv A$$
$$A \wedge \top \equiv A$$
$$A \vee \top \equiv \top$$

## 2.6 ≡ as a congruence

≡ is **more than** simply being an **equivalence relation**.

≡ is a **congruence** with respect to $\vee, \wedge, \rightarrow, \neg$ **thought as operations**, so if the operands are thought as

$$\wedge : \mathcal{F}_{\mathcal{L}}^2 / \equiv \rightarrow {}^{\mathcal{F}_{\mathcal{L}}}/_{\equiv}$$

$$([A]_{\equiv}, [B]_{\equiv}) \rightarrow [A \wedge B]_{\equiv}$$

≡ is a congruence with respect to $\wedge$ as defined above.
If $A \equiv B$ and $C \equiv D$ then $A \wedge B \equiv C \wedge D$.

Exercise: verify that ≡ is a congruence with respect to all the connectives $\wedge, \vee, \rightarrow, \neg$ (also $\top, \bot, \leftrightarrow$ even if they're derived so there would technically be no need).

*Proof.* We demonstrate just the case of ≡ being a congruence with respect to $\neg$.
$A \equiv B$ **iff** $\neg A \equiv \neg B$. For all $v : \mathcal{L} \rightarrow \{0, 1\}$, $\tilde{v}(A) = \tilde{v}(B)$ **iff** $T_{\neg}(\tilde{v}(A)) = I_{\neg}(\tilde{v}(B))$ **iff** $\tilde{v}(\neg A) = \tilde{v}(\neg B)$.

□

Note: it is not true that every equivalence relation is a congruence (with respect to some operations).

We've started by considering the set of particular strings $\mathcal{F}_{\mathcal{L}}$, now we have a set of classes of logical equivalence ${}^{\mathcal{F}_{\mathcal{L}}}/_{\equiv}$ and the operator ≡ can be thought as a congruence.

## 2.7  Partial order relation

Let $R$ be a **binary relation** on $S$, $R \subseteq S^2$ such that

- $R$ is reflexive $(s, s) \in R$ for all $s \in S$
- $R$ is **antisymmetric** for all $(s, t) \in R$, if $(s, t) \in R$ and $(t, s) \in R$ then $s = t$
- $R$ is transitive $(s, t) \in R$ and $(t, r) \in R$ then $(s, r) \in R$

Any **relation with these properties is a partial order relation**. The difference between this and an equivalence relation is the antisymmetric property, which states that if two elements are in the relation in a certain order, the opposite ordering can't be in the relation as well.

Symbols: $\leq, \preceq, \sqsubseteq$.

Some examples of partial orderings for the naturals:

$$(\mathbb{N}, \leq)$$

$a \leq b$ if it exists $c \in \mathbb{N}$ such that $a + b = c$.
This relation has a minimum element, which is 0.

$$(\mathbb{N}, \geq)$$

$a \geq b$ if it exists $c \in \mathbb{N}$ such that $a - b = c$.
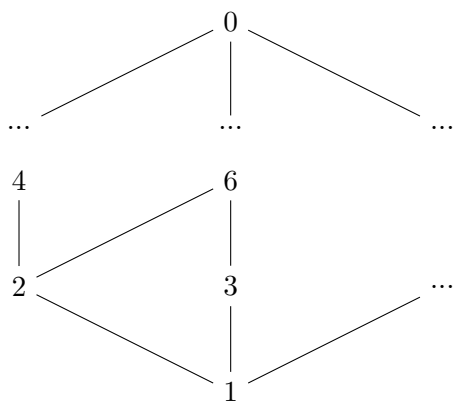In this relation 0 is the maximum element.

$$(\mathbb{N}, =)$$

In this relation every element is on the same level.

Divisibility order:

$$(\mathbb{N}, |)$$

$a|b$ ($a$ divides $b$) if there is $c \in \mathbb{N}$ such that $a \cdot c = b$.



This order has 1 as a minimum, all the prime numbers in the first level (atoms), then all numbers created just by primes and so on until the maximal element, which is zero since there's always a number such that $n \cdot 0 = 0$.

A **set** $P$, **equipped with a partial order relation** $\leq \subseteq P^2$ is called a **partially ordered set**, for short **poset** $(P, \leq)$.

## 2.8 Lattice

To be a **lattice** there must be, **for each element in the poset**, a **greatest lower bound** and a **lowest upper bound**.

Considering $S \subseteq P$, $P$ poset, $(P, \leq)$.

We can define the **infimum** (greatest lower bound): $\inf S$ if it exists is the element $t \in P$ such that for all $z \in P$, if $z \leq s$ for all $s \in S$ then $z \leq t$ ($t$ is the greatest lower bound).

We can also define the **supremum** $\sup S$: if it exists is the element $t \in P$ such that for all $z \in P$, if $s \leq z$ for all $s \in S$ then $t \leq z$ ($t$ is the lowest upper bound).

$$\inf S = \max\{z \in P : z \leq s \text{ for all } s \in S\}$$

$$\sup S = \min\{z \in P : s \leq z; \text{ for all } s \in S\}$$

If they exist.

We can also define:

$$\min S = t \text{ iff } t = \inf S \text{ and } t \in S$$

$$\max S = t \text{ if } t = \sup S \text{ and } t \in S$$

**Example** of a set lacking sup (and then lacking max)

$$S \subseteq \mathbb{Q} \quad S := \{x : x^2 \leq 2\}$$

This has a sup, but not in the rationals. It would be $\sqrt{2} \in \mathbb{R} \setminus \mathbb{Q}$.

While
$$S' \subseteq \mathbb{Q} \quad S' := \{x : x \lneq 1\}$$

Has a sup but not a max, $\sup S' = 1$ but $1 \notin S'$, so $\max S'$ does not exist.

$$S'' \subseteq \mathbb{Q} \quad S'' := \{x : x \leq 1\}$$

$\sup S'' = 1$, $1 \in S''$ so $\max S'' = 1$

**Definition of Lattice (as poset):** A poset $(R, \leq)$ **is a lattice iff** for all $x, y \in R$ there **exists both** $\inf\{x, y\}$ **and** $\sup\{x, y\}$.

Exercise: is our poset a lattice?

$$(\mathbb{N}, \leq)$$

is a lattice because every pair has an greatest lower bound (the smaller number) and a lowest upper bound (the largest number).

$$(\mathbb{N}, |)$$

Divisibility is a lattice, the inf is the greatest common divisor, while the sup is the lowest common multiple.

$$(\mathbb{N}, =)$$

Is NOT a lattice, every element is on the same level, there is no inf or sup.

## 2.9    Algebraic Structure

**(Very sketchy) Definition of Algebraic Structure:** Let $S$ be a set, and $*_1, *_2, \ldots, *_n$ a number of operations (can be infinite) on $S$ (that is functions $*_i : S^{\nu(i)} \to S$, where $\nu(i)$ gives the arity of the operation).

Then $(S, *_1, *_2, \ldots, *_n)$ is an **algebraic structure**, or is an algebra.

By algebra we basically mean a set of operations.

Example:
$$(\mathbb{Z}, +, -, 0)$$
$$+ : \mathbb{Z}^2 \to \mathbb{Z} \quad (a, b) \to a + b$$
$$- : \mathbb{Z}^1 \to \mathbb{Z} \quad a \to -a$$
$$0 : \mathbb{Z}^0 \to \mathbb{Z} \quad 0 \text{ the constant/integer number } 0$$

The arity of these operations is, respectively: $\nu(+) = 2$, $\nu(-) = 1$, $\nu(0) = 0$.

### 2.9.1 Isomorphism

Two **structures** are **isomorphic**

$$(S, *_1, *_2, \ldots, *_n) \cong (R, \odot_1, \odot_2, \ldots, \odot_m)$$

If they mathematically behave in the same manner, i.e. **iff**

$$m = n$$

$$\nu_S^{(i)} = \nu_R^{(i)} \quad \text{for each } i = 1, 2, \ldots, n$$

there exists $f : S \to R$ such as:

$$\begin{cases} f: & \text{injective:} & x \neq y \in S \implies f(x) \neq f(y) \\ f: & \text{surjective:} & \text{for all } y \in R \text{ there is } x \in S \text{ s.th. } f(x) = y \end{cases}$$

$$\implies f \text{ is bijective (1-1 correspondance)}$$

$$f: \text{ is a homomorphism}: f(*_i(s_1, \ldots, s_k)) = \odot_i(f(s_1), f(s_2), \ldots, f(s_k))$$

$$\nu_S(i) = k, \text{ for all } (s_1, \ldots s_k) \in S^k$$

TLDR: The structures are the same, except for their physical nature, they behave the same as far as those operations are concerned.

### 2.9.2 Lattices as algebraic structures

A **lattice as an algebraic structure** is a **set with binary operations**

$$(R, \sqcap, \sqcup) \quad \begin{matrix} \sqcap : R^2 \to R \\ \sqcup : R^2 \to R \end{matrix}$$

such that **both** these **operations** are **associative and commutative**.

$\sqcap, \sqcup$ are related by absorption:

$$\text{for all} \ \ x, y \in R \quad \begin{cases} x \sqcap (x \sqcup y) = x \\ x \sqcup (x \sqcap y) = x \end{cases}$$

A **lattice as a poset** can be **redefined as/associated** to a **lattice as an algebraic structure**

$$(R, \leq) \longleftrightarrow (R, \sqcap, \sqcup)$$

each lattice meant as a poset can be thought of uniquely as an algebra and vice versa.

To do

$$(R, \leq) \rightsquigarrow (R, \sqcap, \sqcup)$$

we just have to define the operations. For all $a, b \in R$ we need to define:

$$a \sqcap b := \inf\{a, b\}$$

$$a \sqcup b := \sup\{a, b\}$$

The rest is an exercise for the reader.

While to do
$$(R, \sqcap, \sqcup) \rightsquigarrow (R, \leq)$$

For all $a, b \in R$, $a \leq b$ iff $a = a \sqcap b$ (or, equivalently, $b = a \sqcup b$).

Check: $a \leq b$ defined as above is such that $\leq$ is

- reflexive: for all $a \in R$, $a \leq a$ iff $a = a \sqcap a$ (by idempotence)
- antisymmetric: ...
- transitive: ...

Doing this we prove that this thing is a poset, but the lattice is a special kind of poset. So now we need to check that the poset $(R, \leq)$ is indeed a lattice. We need to prove that for each pair of element there is an inf and sup. **For each $a, b \in R$ there is** $\inf\{a, b\}$ **and** $\sup\{a, b\}$.

Letting $c \leq a$ and $c \leq b$ we have to show that $c \leq a \sqcap b$, since it means we've proven that $a \sqcap b$ is $\inf\{a, b\}$, so it exists and is defined, $\inf\{a, b\} = a \sqcap b$.

$c = a \sqcap c$, by definition of $\leq$, and from the fact that $c \leq b$ and thus $c = b \sqcap c$
so
$$c = a \sqcap c = a \sqcap (b \sqcap c) = (a \sqcap b) \sqcap c \text{ iff } c \leq a \sqcap b$$

So inf exists and is defined.

The proof that $\sup\{a, b\} = a \sqcup b$ is analogous.

We have proved that $(R, \leq)$ is a lattice (as a poset).

## 2.10 Boolean Algebras

It's an algebra, so a set with some operations and two constants

$$(S, \sqcap, \sqcup, ()^c, 0, 1)$$

The arity of the operations is:

- $\sqcap, \sqcup : S^2 \to S$
- $()^c : S \to S$
- $0, 1 : S^0 \to S$

(The $()^c$ is the complement).

**Additional properties**: $(S, \sqcap, \sqcup)$ is a lattice (has an order), which is

- **Bounded:** for all $a \in S$ there's a minimum $0 \le a$ and a maximum $a \le 1$ such that
$$\begin{cases} a \sqcap 0 = 0 & a \sqcup 1 = 1 \\ a \sqcup 0 = a & a \sqcap 1 = a \end{cases}$$

- **Complemented:** for all $a \in S$, such that $a \sqcap a^c = 0$ and $a \sqcup a^c = 1$.

- **Distributive:** for all $a, b, c \in S$, $a \sqcap (b \sqcup c)$

$$a \sqcap (b \sqcup c) = (a \sqcap b) \sqcup (a \sqcap c)$$
$$a \sqcup (b \sqcap c) = (a \sqcup b) \sqcap (a \sqcup c)$$

So, boolean algebra is a shorthand name for bounded, complemented and distributive lattices.

Examples of boolean algebra: Let $A$ be a set

$$(\mathcal{P}(A), \cap, \cup, \overline{()}, \emptyset, A)$$

Let $\mathcal{P}(A)$ be the powerset of $A$ (set of all subsets of $A$).
Check that this is a boolean algebra. It must be commutative, absorption, bounds, ...

If $A$ is finite, say $|A| = n$, then $|\mathcal{P}(A)| = 2^n$. $2^A$ is a commonly used notation for the powerset.

Every finite Boolean algebra has $2^k$ elements for some $k \in \mathbb{N}$.

**Every finite boolean algebra** can be conceived/is **isomorphic with** the **powerset of some finite set**.

$(\mathcal{P}(A), \subseteq)$, $S, T$, $S \subseteq T$ iff $S = S \cap T$ (or $T = S \cup T$). This is the lattice isomorphic to the boolean algebra defined above.

Example:
$$A = \{a, b\}, \; |\mathcal{P}(A)| = |2^A| = 2^2 = 4$$
$$2^A = \{\emptyset, \{a\}, \{b\}, \{a, b\}\}$$

So
$$(2^A, \subseteq)$$

is a lattice.

**Boolean algebra of truth-values:**

$$(\{0,1\}, I_\wedge = \min, I_\vee = \max, I_\neg = 1-, 0 = I_\bot, 1 = I_\top)$$

This is the **smallest significant boolean algebra** and in the ordering we have that 0 is below 1.

It's equivalent to:
$$\cong (\mathcal{P}(\{*\}), \cap, \cup, \overline{()}, \emptyset, \{*\})$$

**Boolean algebras of $n$-tuples of truth-values**

$$(\{0,1\}^n, \overline{\min}, \overline{\max}, \overline{1-}, (0,0,....,0), (1,1,...,1))$$

Operations are defined pointwise (componentwise):

$$\overline{\min}((b_1, ..., b_n), (c_1, ... c_n)) := (\min\{b_1, c_1\}, ..., \min\{b_n, c_n\})$$

$$\overline{\max}((b_1, ..., b_n), (c_1, ... c_n)) := (\max\{b_1, c_1\}, ..., \max\{b_n, c_n\})$$

$$\overline{1 - (b_1, ..., b_n)} = (1 - b_1, ..., 1 - b_n)$$

### 2.10.1  Lindembaum Algebras of (equivalence classes of) formulas

It is a boolean algebra.

$$(\mathcal{F}_\mathcal{L}/\equiv, \wedge, \vee, \neg, [\bot]_\equiv, [\top]_\equiv)$$

Since $\equiv$ is a congruence with respect to $\wedge, \vee, \neg$ the following **definition is well given** (does not depend on the representatives of the classes).

For all $A, B \in \mathcal{F}_\mathcal{L}$:

$$[A]_\equiv \wedge [B]_\equiv := [A \wedge B]_\equiv$$

$$[A]_\equiv \vee [B]_\equiv := [A \vee B]_\equiv$$

$$\neg[A]_\equiv := [\neg A]_\equiv$$

It has an **order relation** since it's also a **lattice**:

$$(\mathcal{F}_\mathcal{L}/\equiv, \leq) \quad [A]_\equiv \leq [B]_\equiv \ \text{ iff } \ [A]_\equiv = [A \wedge B]_\equiv \text{ or } [B]_\equiv = [A \vee B]_\equiv$$

Some combinatorics about $\mathcal{F}_\mathcal{L}/\equiv$

$$|\mathcal{F}_\mathcal{L}| = |\mathcal{L}| = |\mathbb{N}| = \aleph_0$$

$$|\mathcal{F}_\mathcal{L}/\equiv| = \aleph_0$$

No powerset has cardinality $\aleph_0$, so $|\mathcal{F}_\mathcal{L}/\equiv|$ is **not isomorphic** with any powerset algebra.

By $\mathcal{F}_\mathcal{L}^{(n)}$ we mean the set of all formulas built using only the first $n$ propositional letters $\{p_1, p_2, ..., p_n\}$. Check that

$$|FL^{(n)}/\equiv|$$

is a boolean algebra (called Lindenbaum algebra over the first $n$ letters).

And he checks that, but I'm not paid enough to write it.