Logic for knowledge representation, learning, and inference

Luciano Serafini

Contents

CHAPTER 1

Propositional Logic

1. What is a proposition

According to Stanford Encyclopedia of Philosophy[1] , the term "proposition" has a broad use in contemporary philosophy. It is used to refer to some or all of the following: the primary bearers of truth-value, the objects of belief and other "propositional attitudes" (i.e., what is believed, doubted, known, etc.), the referents of that-clauses, and the meanings of sentences.

Propositions can be expressed by sentences of a natural language (English, Italia, etc.) Examples of propositions are "John is a teacher", "John is rich' and "John is a rock singer". Notice that there is a difference between "sentence" and "proposition". A sentence is a string that *expresses a proposition*. I.e., the meaning of a sentence is a proposition. In natural langue we say that two sentences are equivalent, when they expresses the same proposition. E.g.,

(1) the brother of my mam is blond;
(2) my uncle is blond;

One of the most important characteristic of proposition is that they can take a truth value. IN classical propositional logic, there are only two truth values *true* and *false*. Notice that, one can imagine a situation in which a proposition is not completely true or completely false, or it is both true and false. To treat these situations there are other logics (we will see fuzzy logic). So, don't believe that the world of logic is limited to "true and false".

Complex propositions can be build by combining simpler (atomic) propositions. For instance the sentence "Paolo is painting and Renzo is playing piano" expresses a proposition which is the *conjunction* of the propositions expressed by the sentences "Paola is painting" and "Pietro is playing piano". There are other ways to compose complex proposition from simpler ones. For instance one could "disjunct" two propositions. E.g., the proposition "Paolo is painting or Renzo is playing piano" is the *disjunction* of the two simpler propositions. Another example is negation. E.g., the proposition rexpressed by the sentence "Renzo is not playing the piano" is the result of negating the proposition expressed by the sentence "Renzo is playing the piano".

Notice that the truth value of the simpler proposition determine the truth value of the complex proposition. This property is sometimes referred as *truth-functionality* and has been formally characterised by the Polish logician Alfred Tarski in 1933.

[1] see `https://plato.stanford.edu/entries/propositions/`

2. The language of propositional logic

The language of propositional logic allows to "speak about" propositions. The basic components of a propositional language is the set of *propositional variables* $\mathcal{P}$. This can be a finite or an infinite set (indeed the set of all propositions in general might be infinite).

$$\mathcal{P} = \{p_1, p_2, p_3, \dots\}$$

When we stud When we want to use propos y the general propoerties of PL, we don't care about which proposition is denoted by each single p_i in $\mathcal{P}$; we only assume that each p_i refers to some proposition. This is the reason why p_i is called *variable*. Instead, when we want to use PL to represent some concrete scenario, we have to specify the specific proposition which is denoted by each propositional variable. For instance if we want to use PL to describe what Paola and Renzo are doing. we can define the set of propositional variables:

$$\mathcal{P} = \{p, r\}$$

where

- p is the proposition expressed by the sentence "Paola is painting";
- r is the proposition expressed by the sentence "Renzo is playing the piano".

The language of PL allows to express also complex proposition which are the result of negating a proposition, and combining two propositions with conjunction, disjunction, implication, and equivalence. To allow this PL introduces the following set of symbols called *propositional connectives*

- $\neg$ for negation (not);
- $\wedge$ for cojunction (and);
- $\vee$ for disjunction (or);
- $\rightarrow$ for implication (if ... then ...);
- $\leftrightarrow$ for equivalence (if and only if).

In the literature one can find alternative symbols for implication and equivalence, which are $\supset$ and $\equiv$. We will use both the symbols. Two additional symbols "(" and ")" are also added to PL, which allows one to express the correct separation of simple propositions that occour in a complex propositions. Parenthesis in PL play the similar role played by punctuation symbols (e.g., ".", ";", ",", ...) in natural language.

In summary, the *alphabet* of a propositional language is composed by

- a set $\mathcal{P}$ of propositional variables (also called non-logical symbols);
- the set $\neg$, $\wedge$,
  vee, $\rightarrow$, and $\leftrightarrow$ of propositional connectives (also called logical symbols);
- parenthesis symbols (and).

From this alphabet the set of *well formed formula* are defined by induction as follows:

Definition 1.1 (Well formed formulas). *Given a set $\mathcal{P}$ of propositional variables, the set of* well formed formulas *is inductively defined as follows:*

(1) every $p \in \mathcal{P}$ is a well formed formula;
(2) if ϕ is a well formed formula then $\neg\phi$ is a well formed formula;

(3) if ϕ and ψ are well formed formula, then $\phi \circ \psi$ is a well formed formula for $\circ \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$;
(4) nothing else is a well formed formula.

Each element $p \in \mathcal{P}$ is called also atomic formula*. The set* wffs($\mathcal{P}$) *is the set of well formed formulas that contains only the propositions in $\mathcal{P}$*

An additional symbol, that is often use in propositional logic is the $\perp$ symbol. Intuitively $\perp$ is a constant that represents the proposition that is always false.

The term "formula" us used as a short version of "well formed formulas" when there is no confusion. We also shorten well formed formula with "wff".

Example 1.1 (Formulas and non formulas).

Formulas	**Non formulas**
$p \rightarrow q$	pq
$p \rightarrow (q \rightarrow r)$	$(p \rightarrow \wedge((q \rightarrow r)$
$(p \wedge q) \rightarrow r$	$p \wedge q \rightarrow \neg r \neg$

Notice that the formulas on the left can be build by applying the rules of Definition 1.1. For instance the formula $(p \wedge q) \rightarrow r$ can be buld as follows

(1) p is a wff for item 1;
(2) q is a wff for item 1;
(3) $p \wedge q$ is a wff for item 3;
(4) $(pwedgeq) \rightarrow r$ is a wff for item 3.

Notice that in the last step we use parenthesis, in order to specify how the formula is build. Indeed if one does not use parenthesis, the formula would look as

$$p \wedge q \rightarrow r$$

The above formula however can be build in two ways.

- *first build $p \wedge q$ and then $p \wedge q \rightarrow r$ (as we indeed do)*
- *first buld $q \rightarrow r$ and then $p \wedge q \rightarrow r$*

To distinguish the two ways of constructing the formula we use parentes. obtaining

$$(p \wedge q) \rightarrow r \tag{1}$$

$$p \wedge (q \rightarrow r) \tag{2}$$

Notice that the two formulas above represents different propositions.

Example 1.2. *Suppose that*

- *p stands for "Paola is painting";*
- *q stands for "outside it's raining";*
- *r stands for "Renzo is playing the piano".*

we have that the two formulas refers to the propositions expressed by

(3)
If Paola is painting and outside it is raining, then Renzo is playing the piano;

(4)
Paola is painting and, if outside it is raining, then Renzo is playing the piano

Notice that the proposition expressed by the two sentences above are different. So they are the corresponding propositional formula.

In some cases, the use of parenthesis will result in an awkward notation. In order to minimize the usage of parenthesis only when they are necessary, PL provide a "default" order of construction in case there are no parenthesis. This default order of application of rule of construction is specified by associating a *priority ordering* between connectives. $\neg$ has the highest priority, then $\wedge$, $\vee$, $\rightarrow$ and $\equiv$.

Symbol	Priority
$\neg$	1
$\wedge$	2
$\vee$	3
$\rightarrow$	4
$\leftrightarrow$	5

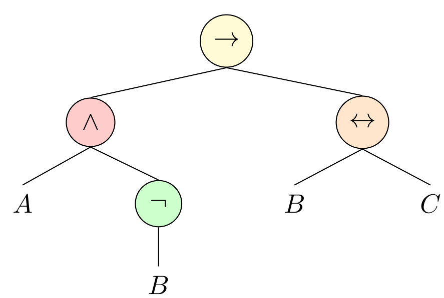
In absence of parenthesis the above priorities are applied. Therefore the formula $p \wedge q \rightarrow r$ is considered to be $(p \wedge q) \rightarrow r$. If we want to refer to the other formula we have to explicitly add parenthesis.

A formula can be seen as a tree. The leaves of the tree are propositional variables contained in the formula and the intermediate (non-leaf) nodes are associated to connectives that are used in order to build the complex formula.

EXAMPLE 1.3 (Tree of the formula). *The tree of the formula*

$$A \wedge \neg B \rightarrow (B \leftrightarrow C)$$

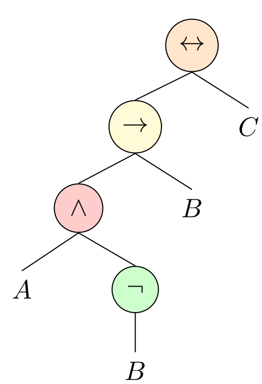
is the following:



Notice that, in order to force that the $\leftrightarrow$ *connective should be applied before the* $\rightarrow$ *connective we have to use parenthesis. Indeed the tree of the formula without parenthesis, i.e.,*

$$A \wedge \neg B \rightarrow B \leftrightarrow C$$

is the following:



Informally, a subformula is a part of a wff which is itself a wff. More formally the best approach is via the idea of the constructional history for a wff, with the subformulae being the wffs that appear in the history. Any decent textbook will explain this, and explain how you set out constructional histories as trees.

The set of formulas which need to be constructed in order to build a complex formula ϕ are called the *subformulas of* ϕ. The trees of the subformulas of a formula ϕ are the sub-trees of the tree of ϕ. A formal definition of sub-formula of ϕ is the following:

DEFINITION 1.2.

- *A is a* subformula *of itself*
- *A and B are* subformulas *of $A \wedge B$, $A \vee B$ $A \supset B$, e $A \equiv B$*
- *A is a subformula of $\neg A$*
- *if A is a subformula of B and B is a subformula of C, then A is a subformula of C.*
- *A is a* proper subformula *of B if A is a subformula of B and A is different from B.*

EXAMPLE 1.4. *The subformulas of $(p_0 \vee \neg p_1) \to (p_2 \wedge p_1)$ are: $(p_0 \vee \neg p_1) \to (p_2 \wedge p_1)$, $p_0 \wedge \neg p_1$, $p_2 \wedge p_1$, p_0, $\neg p_1$, p_2 and p_1.*

NOTATION 1.1. *Unfortunately, books on mathematical logic use widely varying notation for the Boolean operators; furthermore, the operators appear in programming languages with a different notation from that used in mathematics textbooks. The following table shows some of these alternate notations.*

Connective	*Alternative*	*programming languages*
$\neg$	$\sim$	~ ! -
$\wedge$	$\&$	& && ^
$\vee$	$\Vert$	| ||
$\to$	$\supset$ $\Rightarrow$	=>, ->
$\leftrightarrow$	$\Leftrightarrow$ $\equiv$	<=>, <->, = %

3. Interpretation of propositional formulas

Informally speaking an interpretation of a propositional language represents a state of affairs that allows one to establish for every proposition if it *true* or *false*, or equivalently if it *holds* or it *does not hold*. So for instance if we have the set of $\mathcal{P}$ are $\{p, q, r\}$ and they denote the propositions as described in Example 1.2, an interpretation correspond to a specific situation in which for instance Paola is painting, outside it is not raining, and Renzo is not playing piano. Since in PL we have that the truth value of complex propositions is fully determined by the truth value of the simplest (aka atomic) propositions, an interpretation can be specified by saying if p is holds or does not hold for every $p \in \mathcal{P}$. Let's now define this notion formally:

DEFINITION 1.3. *An* interpretation *of a propositional language on the set of propositional variables $\mathcal{P}$, is a function $\mathcal{I} : \mathcal{P} \to \{True, False\}$.*

Alternative notation for *True* and *False* that we will unse in this notes and are used in other books are 0 and 1, $\top$ and $\bot$. An alternative and equivalent definition of interpretation is the following

Definition 1.4. *An* interpretation *of the set of propositional variables $\mathcal{P}$ is any subset $\mathcal{I} \subseteq \mathcal{P}$.*

The two definitions are equivalent under the correspondence

$$\mathcal{I}(p) = \textit{True} \text{ if and only if } p \in \mathcal{I}\mathcal{I}(p) = \textit{False} \text{ if and only if } p \notin \mathcal{I}$$

We will use alternatively both the definition depending on convenience. Notice that if $\mathcal{P}$ contains n propositional variables, (i.e, $|\mathcal{P}| = n$) then there are 2^n distinct interpretations. This corresponds to the fact that a set $\mathcal{P}$ that contains n elements has 2^n distinct subsets.

Example 1.5. *Suppose that the set $\mathcal{P}$ of propositional variaables is equal to $\{p, q, r\}$, then there are $2^3 = 8$ propositional interpretations of $\mathcal{P}$. The following table reports all of them in the functional form $\mathcal{I} : \mathcal{P} \rightarrow \{\textit{True}, \textit{False}\}$ and in the setwise form $\mathcal{I} \subseteq \mathcal{I}$.*

	Functional form			*Set theoretic form*
	p	q	r	
$\mathcal{I}_1$	*True*	*True*	*True*	$\{p, q, r\}$
$\mathcal{I}_2$	*True*	*True*	*False*	$\{p, q\}$
$\mathcal{I}_3$	*True*	*False*	*True*	$\{p, r\}$
$\mathcal{I}_4$	*True*	*False*	*False*	$\{p\}$
$\mathcal{I}_5$	*False*	*True*	*True*	$\{q, r\}$
$\mathcal{I}_6$	*False*	*True*	*False*	$\{q\}$
$\mathcal{I}_7$	*False*	*False*	*True*	$\{r\}$
$\mathcal{I}_8$	*False*	*False*	*False*	$\{\}$

The truth value for propositional letters assigned by an interpretation $\mathcal{I}$ one can define when a formula is true (or holds) in the interpretation $\mathcal{I}$.

Definition 1.5 ($\mathcal{I}$ satisfies a formula A, $\mathcal{I} \models A$). *A formula A is* satisfied by *an interpretation $\mathcal{I}$, in symbols*

$$\mathcal{I} \models A$$

according to the following inductive definition:

- *If $p \in \mathcal{P}$, $\mathcal{I} \models p$ if $\mathcal{I}(p) = \textit{True}$;*
- *$\mathcal{I} \models \neg A$ if not $\mathcal{I} \models A$ (also written $\mathcal{I} \not\models A$);*
- *$\mathcal{I} \models A \wedge B$ if, $\mathcal{I} \models A$ and $\mathcal{I} \models B$;*
- *$\mathcal{I} \models A \vee B$ if $\mathcal{I} \models A$ or $\mathcal{I} \models B$;*
- *$\mathcal{I} \models A \rightarrow B$ if either $\mathcal{I} \not\models A$ or $\mathcal{I} \models B$;*
- *$\mathcal{I} \models A \leftrightarrow B$ if $\mathcal{I} \models A$ iff $\mathcal{I} \models B$.*

If $\mathcal{I}$ is an interpretation and A a formula, following expressions has the same meaning

- $\mathcal{I} \models A$;
- $\mathcal{I}$ satisfies A;
- A is true in $\mathcal{I}$;
- A holds in $\mathcal{I}$;
- A is satisfied by $\mathcal{I}$;
- $\mathcal{I}(A) = \textit{True}$.

Notice that, if we have to check if a formula A is true in an interpretation $\mathcal{I}$, we have to take into account only the assignments that $\mathcal{I}$ does to the propositional variables that occours in A. This fact isi reflected by the following property:

PROPOSITION 1. *For every pair of interpretations $\mathcal{I}$ and $\mathcal{I}'$, if $\mathcal{I}(p) = \mathcal{I}'(p)$ for all the propositional variables p of a formula A, then $\mathcal{I} \models A$ iff $\mathcal{I}' \models A$*

In other words if $\mathcal{I}$ and $\mathcal{I}'$ differs only on the propositional variables that do not appear in A, then the truth values of A in $\mathcal{I}$ and $\mathcal{I}'$ are equal. As a consequence, to check if $\mathcal{I} \models A$ it is enough to consider the truth values that $\mathcal{I}$ assigns to the propositional variables appearing in A.

3.1. Material Implication. Some discussion is necessary about the semantics of implication $\rightarrow$. The operator $\rightarrow$ is called *material implication*; p is the *antecedent* and q is the *consequent*. Material implication does not claim causality; that is, it does not assert there the antecedent causes the consequent (or is even related to the consequent in any way). A material implication merely states that if the antecedent is true the consequent must be true; so it can be falsified only if the antecedent is true and the consequent is false. This definition of the conditions under which $a \rightarrow b$ is true is easily acceptable when a (the premise of the implication) is true, but when a is false, according to the definition, we have that $a \rightarrow b$ is true. In other words according to the condition given above we have that

$$\mathcal{I} \models a \rightarrow b \text{ if and only if } \mathcal{I} \models \neg a \vee b$$

so in other words $a \rightarrow b$ and $\neg a \vee b$ are equivalent propositions. This is sometimes very unintuitive. Consider for instance the statement:

the moon is made of cheese implies that the earth is flat

which can be represented by the propositional formula

$$c \rightarrow f$$

with c representing the proposition "the moon is made of cheese" and f the proposition "the earth is flat". According to the formal sematnics we have that $c \rightarrow f$ is true in the current state of the world (as we know that the premise is false), however the sentence in natural language suggests some cauation between the premise c and the conclusion f, which is not formalized by the implication. Other paradixical formulas are the following. They are true in all interpretation, however they are not intuitively acceptable:

- $(\neg p \wedge p) \rightarrow q$: p and its negation imply q. This is the paradox of entailment.
- $p \rightarrow (q \rightarrow p)$: if p is true then it is implied by every q.
- $\neg p \rightarrow (p \rightarrow q)$: if p is false then it implies every q. This is referred to as 'explosion'. In these cases, the statement $p \rightarrow q$ is said to be vacuously true.
- $p \rightarrow (q \vee \neg q)$: either q or its negation is true, so their disjunction is implied by every p.
- $(p \rightarrow q) \vee (q \rightarrow r)$: if p, q and r are three arbitrary propositions, then either p implies q or q implies r. This is because if q is true then p implies it, and if it is false then q implies any other statement. Since r can be p, it follows that given two arbitrary propositions, one must imply the other, even if they are mutually contradictory. For instance, "Nadia is in Barcelona implies Nadia is in Madrid, or Nadia is in Madrid implies Nadia is in Barcelona."is always true. This sounds like nonsense in ordinary discourse.

- $\neg(p \to q) \to (p \wedge \neg q)$: if p does not imply q then p is true and q is false. N.B. if p were false then it would imply q, so p is true. If q were also true then p would imply q, hence q is false. This paradox is particularly surprising because it tells us that if one proposition does not imply another then the first is true and the second false.

Suggestion: As a practice in order to avoid these types of paradoxes it is better to read $a \to b$ directly as $\neg a \vee b$.

3.2. The low of excluded middle. The formula $p \vee \neg p$ is valid (or equivalently it is a tautology). Indeed independently from the truth value assigned to p by an interpretation $\mathcal{I}$, we have that $\mathcal{I} \models p \vee \neg p$.

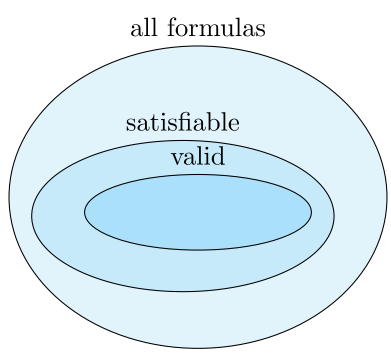
This tautology, called the *law of excluded middle*, is a direct consequence of our basic assumption that a proposition is a statement that is either true or false. Thus, the logic we will discuss here, so-called Aristotelian logic, might be described as a “2-valued” logic, and it is the logical basis for most of the theory of modern mathematics, at least as it has developed in western culture. There is, however, a consistent logical system, known as constructivist, or intuitionistic, logic which does not assume the law of excluded middle. This results in a “3-valued” logic in which one allows for a third possibility, namely,”other”. In this system proving that a statement is “not true” is not the same as proving that it is “false”, so that indirect proofs, which we shall soon discuss, would not be valid. If you are tempted to dismiss this concept you should be aware that there are those who believe that in many ways this type of logic is much closer to the logic used in computer science than Aristotelian logic. You are encouraged to explore this idea: there is plenty of material to be found in your library or through the worldwide web.

3.3. Valid, Satisfiable, and unsatisfiable formulas.

Definition 1.6. *A formula A is*

- Valid *if* for all interpretations $\mathcal{I}$, $\mathcal{I} \models A$. *The notation $\models A$ (with no interpretation in the front) stands for “A is valid”;*
- Satisfiable *if* there is an interpretations $\mathcal{I}$ *s.t.,* $\mathcal{I} \models A$
- Unsatisfiable *if* for no interpretations $\mathcal{I}$, $\mathcal{I} \models A$

Validity, satisfiable, and unsatisfiability are not independent concepts, they are related one another. This relation is highlighted in the following diagram



Stating that

$$\text{Valid formulas} \subset \text{Satisfiable formulas} \subset \text{Well formed formulas}$$

This implies that if a formula is valid it is also satisfiable, and if a formula it is unsatisfiable (not satisfiable) it is also not valid. Alternative terms for valid, satisfiable and unsatisfiable formula are the following:

- *tautology* is a synonym of *valid*;
- *contingency* is a synonym of *satisfiable*;
- *contradiction* is a synonym of *unsatisfiable*.

EXAMPLE 1.6.

$$\begin{array}{l}
\textit{Satisfiable}\left\{\begin{array}{c} A \to A \\ A \vee \neg A \\ \neg\neg A \equiv A \\ \neg(A \wedge \neg A) \\ A \wedge B \to A \\ A \to A \vee B \\ p \vee q \\ p \to q \\ \neg(p \vee q) \to r \end{array}\right. \quad \left.\begin{array}{c} A \to A \\ A \vee \neg A \\ \neg\neg A \equiv A \\ \neg(A \wedge \neg A) \\ A \wedge B \to A \\ A \to A \vee B \end{array}\right\}\textit{Valid} \\
\textit{Uatisfiable}\left\{\begin{array}{c} A \wedge \neg A \\ \neg(A \to A) \\ A \equiv \neg A \\ \neg(A \equiv A) \end{array}\right. \quad \left.\begin{array}{c} p \vee q \\ p \to q \\ \neg(p \vee q) \to r \\ A \wedge \neg A \\ \neg(A \to A) \\ A \equiv \neg A \\ \neg(A \equiv A) \end{array}\right\}\textit{Non Valid}
\end{array}$$

Prove that the blue formulas are valid, that the magenta formulas are satisfiable but not valid, and that the red formulas are unsatisfiable.

In the following we provide the proof for some formulas and leave the others by exercize.

- *$A \to A$ is valid. By Definition 1.5, $\mathcal{I} \models A \to A$ if and only if either $\mathcal{I} \models A$ or $\mathcal{I} \not\models A$, which holds for every $\mathcal{I}$.*
- *$A \vee \neg A$ is valid. By Definition 1.5, $\mathcal{I} \models A \vee \neg A$ if and only if $\mathcal{I} \models A$ or $\mathcal{I} \models \neg A$. Furhtermore, by Definition 1.5, $\mathcal{I} \models \neg A$ if and only if $\mathcal{I} \not\models A$. So $\mathcal{I} \models A \vee \neg A$ iff $\mathcal{I} \models A$ or $\mathcal{I} \not\models A$, which is true for every $\mathcal{I}$.*
- *$\neg\neg A \equiv A$ is valid. Suppose by contradiction that there is an $\mathcal{I}$ such that $\mathcal{I} \models \neg\neg A$ and $\mathcal{I} \not\models A$. By Definition 1.5, If $\mathcal{I} \models \neg\neg A$ then $\mathcal{I} \not\models \neg A$. Furtherore, again by Definition 1.5, $\mathcal{I} \not\models A$ then $\mathcal{I} \models \neg A$. So by assuming that there is an interpretation $\mathcal{I}$ that satisfies $\neg\neg A$ and does not satisfy A we reach a contraddiction that $\mathcal{I} \models \neg A$ and $\mathcal{I} \not\models \neg A$. Therefore there is no such an $\mathcal{I}$. This implies that for all interpretations $\mathcal{I} \models \neg\neg A$ if and obnly if $\mathcal{I} \models A$, and due to Definition 1.5, $\mathcal{I} \models \neg\neg A \equiv a$ for all $\mathcal{I}$.*
- *$p \vee q$ is satisfiable. To show that a formula is satisfiable it is sufficient to find an interpretation that makes it true. Let $\mathcal{I}$ be such that $\mathcal{I}(p) = True$, then (independently on the interpretation of q) we have that according to Definition 1.5 $\mathcal{I} \models p \vee q$.*
- *$p \vee q$ is not valied. To show that a formula is not valid we have to find an interpretation that does not satisfy it. Consider the interpretation $\mathcal{I}$ with $\mathcal{I}(p) = False$ and $\mathcal{I}(q) - False$. We have that by Definitin 1.5 $\mathcal{I} \not\models p \vee q$.*
- *$A \wedge \neg A$ is unsatisfiable. We have to prove that for all interpretation s $\mathcal{I}$, $\mathcal{I} \not\models A \wedge \neg A$. Suppose by contraddiction that there is an interpretation such that $\mathcal{I} \models A|wedge\neg A$, then by Definition 1.5 $\mathcal{I} \models A$ and $\mathcal{I} \models \neg A$. The last fact implies that $\mathcal{I} \not\models A$. Since by assuming that there is an $\mathcal{I}$ that satisfies $A \wedge \neg A$ we reach a contraddiction that $\mathcal{I} \models A$ and $\mathcal{I} \models \neg A$,*

we can conclude that there is no such $\mathcal{I}$, and therefore that $A \wedge \neg A$ is unsatisfiable.

Definition 1.7 (Models of a formula). *For every formula A the set* models(A), the models of A, *is the set* $\{\mathcal{I} \mid \mathcal{I} \models A\}$, *i.e., the set of truth assignments (interpretations) to the propositional variables* prop(A) *that satisfy A;*

The sef of models of a complex formula can be computed in terms of the models of its direct subformulas. However, in computing the models of the subformulas we have to take into account the assignments to the proposition of the entire formulas. For this reason we introduce a generalization of the above definition

Definition 1.8. *If $\mathcal{P}$ is a set of propositional variables that contains* prop(A) *then* $\text{models}_{\mathcal{P}}(A)$ *is the set of assignments to $\mathcal{P}$ that satisfies A.*

Notice that the following facts holds

- if A is satisfiable $\text{models}(A) \neq \emptyset$;
- if A is valide $\text{models}(A) = 2^{\text{prop}(A)}$, i.e., the set of all interpretations of prop(A);
- if A is unsatisfiable then $\text{models}(A) = \emptyset$.
- $\text{models}(\neg A) = 2^{\text{prop}(A)} \setminus \text{models}(A)$;
- $\text{models}(A \wedge B) = \text{models}_{\text{prop}(A\wedge B)}(A) \cap \text{models}_{\text{prop}(A\wedge B)}(B)$;
- $\text{models}(A \vee B) = \text{models}_{\text{prop}(A\vee B)}(A) \cup \text{models}_{\text{prop}(A\vee B)}(B)$;
- $\text{models}(A \rightarrow B) = \text{models}_{\text{prop}(A\rightarrow B)}(\neg A) \cup \text{models}_{\text{prop}(A\rightarrow B)}(B)$
- $\text{models}(A \equiv B = \text{models}_{\text{prop}(A\equiv B)}(A) \cap \text{models}_{\text{prop}(A\equiv B)}(B) \cup \models (\neg A) \cap \models$

Validity, satisfiability, and unsatisfiability of a formula A is also related to validity, satisfiability and unsatisfiability of its negation, i,e. $\neg A$.

Proposition 2.

(1) A is valid if and only if $\neg A$ is unsatisfiable;
(2) A is satisfiable if and only if $\neg A$ is not valid;
(3) A is not valid if $\neg A$ is satisfiable;
(4) A is unsatisfiable if $\neg A$ is valid.

Proof. We prove the first two points and left the other two by exercize.

(1) A is valid then $\text{models}(A) = 2^{\mathcal{P}}$. Since $\text{models}(\neg A) = 2^{\mathcal{P}} \setminus \text{models}(A)$ we have that $\text{models}(\neg A) = \emptyset$, and therefore $\neg A$ is unsatisfiable.
(2) A is satisfiable then $\text{models}(A) \neq \emptyset$. and therefore $\text{models}(\neg A) = 2^{\mathcal{P}} \setminus \text{models}(A) \neq 2^{\mathcal{P}}$ which means that $\neg A$ is not valid.

$\square$

The definition of satisfiability, validity, and unsatisfiability can be extended also to sets of formulas as follows:

Definition 1.9. *A set of formulas Γ is*

- Valid *if for all interpretations $\mathcal{I}$, $\mathcal{I} \models A$* for all formulas $A \in \Gamma$
- Satisfiable *if there is an interpretations $\mathcal{I}$, $\mathcal{I} \models A$* for all $A \in \Gamma$
- Unsatisfiable *if for no interpretations $\mathcal{I}$,, s.t. $\mathcal{I} \models A$* for all $A \in \Gamma$

Proposition 3. *For any* finite set *of formulas Γ, (i.e., $\Gamma = \{A_1, \ldots, A_n\}$ for some $n \geq 1$), Γ is valid (resp. satisfiable and unsatisfiable) if and only if $A_1 \wedge \cdots \wedge A_n$ is valid (resp, satisfiable and unsatisfiable).*

We leave the proof of the previous proposition by exercize.

4. Logical consequence

Logical consequence is one of the key notion of every logic. It is the base of correct inference. Intuitively a proposition, called consequence, logically follows from (or equivalently, is a logical consequence of) a set of propositions, called premises or hypothesis, if from the assumption that the hypothesis are true we can conclude with cartainty that the consequence is also true. For instance if $x > y$ and $y > z$ we can conclude that $x - z > 0$. Logical consequence is strictly connected with the definition of truth of a formula in an interpretation. One of the main motivation of using logic is to make rigorous what is a "valid argument", i.e., when one fact follows form some other facts. Intuitively a "valid argument" is the one that derive true facts from true facts. To this aim, we use the notion of *logical consequence* (Some books refer to logical implication and entailment.)

DEFINITION 1.10. *A formula A is a logical consequnece of a set of formulas Γ, in symbols $\Gamma \models A$, if for all interpretations $\mathcal{I}$, if $\mathcal{I} \models C$ for all $C \in \Gamma$ then $\mathcal{I} \models B$. If Γ contains is a finite set of formulas $\{A_1, \dots, A_n\}$, then we use the notation*

$$A_1, \dots, A_n \models A$$

to denote that A is a logical consequence of Γ.

In other words, $\Gamma \models A$ means that whenever all the wffs in Γ are true, then A must also be true. Notice that this definition does not say anything about interpretations under which one or more of the wffs in Γ are false. In this case, we don't care whether A is true or false.

EXAMPLE 1.7. *q is a logical consequence of $\{p \to q, \neg p \to q\}$. In symbols $p \to q, \neg p \to q \models q$. To check this, since we have to look to all the interpretations we can build a need a truth table.*

	p	q	$p \to q$	$\neg p \to q$
$\mathcal{I}_1$	*True*	*True*	*True*	*True*
$\mathcal{I}_2$	*True*	*False*	*False*	*True*
$\mathcal{I}_3$	*False*	*True*	*True*	*True*
$\mathcal{I}_4$	*False*	*False*	*True*	*False*

$p \to q, \neg p \to q \models q$ holds since all the interpretations that satisfy $p \to q$ and $\neg p \to \neg q$ (i.e.,$\mathcal{I}_1$ and $\mathcal{I}_3$) also satisfy q.

NOTATION 1.2. *Given a non empty finite set of formulas $\Gamma = \{A_1, \dots, A_n\}$*

$$\bigwedge \Gamma \text{ or equivalently } \bigwedge_{i=1}^{n} A_i$$

which stands for $A_1 \wedge A_2 \wedge \cdots \wedge A_n$

$$\bigvee \Gamma \text{ or equivalently } \bigvee_{i=1}^{n} A_i$$

which stands for $A_1 \vee A_2 \vee \cdots \vee A_n$

When Γ is empty we extend the notation as follows

$$\bigwedge \Gamma \text{ denotes } \top$$

and

$$\bigvee \Gamma \text{ denotes } \bot$$

4.1. Properties of propositional logical consequence.

PROPOSITION 4. *If* Γ *and* Σ *are two sets of propositional formulas and* A *and* B *two formulas, then the following properties hold:*

Reflexivity:: $\{A\} \models A$
Monotonicity:: *If* $\Gamma \models A$ *then* $\Gamma \cup \Sigma \models A$
Cut:: *If* $\Gamma \models A$ *and* $\Sigma \cup \{A\} \models B$ *then* $\Gamma \cup \Sigma \models B$
Deduction theorem:: *If* $\Gamma, A \models B$ *then* $\Gamma \models A \to B$
Refutation principle": $\Gamma \models A$ *iff* $\Gamma \cup \{\neg A\}$ *is unsatisfiable*

PROOF.

Reflexifity: For all $\mathcal{I}$ if $\mathcal{I} \models \{A\}$ then $\mathcal{I} \models A$.

Monotonicity: For all $\mathcal{I}$ if $\mathcal{I} \models \Gamma \cup \Sigma$, then $\mathcal{I} \models \Gamma$, by hypothesis ($\Gamma \models A$) we can infer that $\mathcal{I} \models A$, and therefore that $\Gamma \cup \Sigma \models A$

Cut:: For all $\mathcal{I}$, if $\mathcal{I} \models \Gamma \cup \Sigma$, then $\mathcal{I} \models \Gamma$ and $\mathcal{I} \models \Sigma$. The hypothesis $\Gamma \models A$ implies that $\mathcal{I} \models A$. Since $\mathcal{I} \models \Sigma$, then $\mathcal{I} \models \Sigma \cup \{A\}$. The hypothesis $\Sigma \cup \{A\} \models B$, implies that $\mathcal{I} \models B$. We can therefore conclude that $\Gamma \cup \Sigma \models B$.

Deduction theorem: Suppose that $\mathcal{I} \models \Gamma$. If $\mathcal{I} \not\models A$, then $\mathcal{I} \models A \to B$. If instead $\mathcal{I} \models A$, then by the hypothesis $\Gamma, A \models B$, implies that $\mathcal{I} \models B$, which implies that $\mathcal{I} \models B$. We can therefore conclude that $\mathcal{I} \models A \to B$.

Refutation principle: ($\Longrightarrow$) Suppose by contradiction that $\Gamma \cup \{\neg A\}$ is satisfiable. This implies that there is an interpretation $\mathcal{I}$ such that $\mathcal{I} \models \Gamma$ and $\mathcal{I} \models \neg A$, i.e., $\mathcal{I} \not\models A$. This contradicts that fact that for all interpretations that satisfies Γ, they satisfy A ($\Longleftarrow$) Let $\mathcal{I} \models \Gamma$, then by the fact that $\Gamma \cup \{\neg A\}$ is unsatisfiable, we have that $\mathcal{I} \not\models \neg A$, and therefore $\mathcal{I} \models A$. We can conclude that $\Gamma \models A$.

□

The above property has an important impact in using propositinal logic for representing the knowledge of an artificial agent. In particular the *monotonicity* property, states that by adding new knowledge you never "delete" the old knowledge. For instance if an agent represent the fact that all birds flies, with the implication $\mathsf{bird} \to \mathsf{flies}$, and the fact that a penguin is a bird with the implication $\mathsf{penguin} \to \mathsf{bird}$, then this automatically implies that $\mathsf{penguin} \to \mathsf{flies}$. But we know that penguins do not fly, Humans adjust this problem by providing the additional knowledge that penguins are exceptions and therefore they don't fly. This is not possible in propositional logic, since if we add the fact that $\mathsf{penguin} \to \mathsf{exceptionalBird}$, and $\mathsf{exceptionalBird} \to \neg\mathsf{fly}$, we don't delete the fact that a penguin is a bird, and therefore we still derive that penguins fly. In order to cope with this type of representation problem, researchers in knowledge representation in AI introduces "non monotonic" logics Brewka 1989.

The *deduction theorem* states that logical consequence that involves a finite set of formulas can be "internalized" in an implications. Indeed an immediate consequence of the deduction theroem is that $A_1, \dots, A - n \models A$ implies that $A_1 \to (A_2 \to \dots \to (A_{n-1} \to A_n) \dots))$ is a valid formulas.

The *Refutation principle* is also a very important property, that allows to transform the problem of checking logical consequence in to the problem of checking satisfiability of a set of formulas. We will explain this method in the next chapter that is dedicated to algorithms for checking satisfiability.

Another inportan property of the logical consequence in propositional logic is the so called *compactness theorem*n

THEOREM 1.1 (Compactness). *If* $\Gamma \models A$ *then* $\Gamma_0 \models A$ *for some finite subset* $\Gamma_0 \subseteq \Gamma$.

We report one proof of the compactness theorem at the end of this chapter.

5. Logical equivalence

DEFINITION 1.11. *Two formulas A and B are* logically equivalent *if and only if they are true under the same set of interpretations. Alternatively A and B are equivalent if* $[[A]] = [[B]]$.

With an abuse of notation, we write $A \equiv B$ to state that A is logically equivalent to B. Let us now analyse the different type of equivalence formulas we have in PL.

5.0.1. *Absorption of* $\top$ *and* $\bot$. Remember that $\top$ and $\bot$ are usually added to the language of propositional logic, and they are mapped to *True* and *False* respectively by all the interpretations. The appearance of these constants in a formula can collapse the formula so that the binary operator is no longer needed; it can even make a formula become a constant whose truth value no longer depends on the non-constant sub-formula.

$$
\begin{aligned}
A \vee \top &\equiv \top & \qquad A \vee \bot &\equiv A \\
A \wedge \top &\equiv A & A \wedge \bot &\equiv \bot \\
A \rightarrow \top &\equiv \top & A \rightarrow \bot &\equiv \neg A \\
\top \rightarrow A &\equiv A & \bot \rightarrow A &\equiv \top \\
A \leftrightarrow \top &\equiv A & A \leftrightarrow \bot &\equiv \neg A
\end{aligned}
$$

5.0.2. *Identical Operands.* Collapsing can also occur when both operands of an operator are the same or one is the negation of another.

$$
\begin{aligned}
A &\equiv \neg\neg A \\
A \wedge A &\equiv A \\
A \vee A &\equiv A \\
A \wedge \neg A &\equiv \bot \\
A \vee \neg A &\equiv \top \\
A \rightarrow A &\equiv \top \\
A \leftrightarrow A &\equiv \top
\end{aligned}
$$

5.0.3. *Commutativity, Associativity and Distributivity.* The binary Boolean operators are commutative, except for implication.

$$
\begin{aligned}
A \vee B &\equiv B \vee A \\
A \wedge B &\equiv B \wedge A \\
A \leftrightarrow B &\equiv B \leftrightarrow A
\end{aligned}
$$

If negations are added, the direction of an implication can be reversed:

$$A \rightarrow B \equiv \neg B \rightarrow \neg A$$

The formula $\neg B \to \neg A$ is called *the contrapositive* of $A \to B$. Disjunction, conjunction, equivalence are associative.

$$A \vee (B \vee C) \equiv (A \vee B) \vee C$$
$$A \wedge (B \wedge C) \equiv (A \wedge B) \wedge C$$
$$A \leftrightarrow (B \leftrightarrow C) \equiv (A \leftrightarrow B) \leftrightarrow C$$

Disjunction and conjunction distribute over each other

$$A \vee (B \wedge C) \equiv (A \vee B) \wedge (A \vee C)$$
$$A \wedge (B \vee C) \equiv (A \wedge B) \vee (A \wedge C)$$

5.0.4. *De Morgan laws.* Negating a conjunction results in a disjunction of the negated conjuncts, and viceversa

$$\neg(A \wedge B) \equiv \neg A \vee \neg B$$
$$\neg(A \vee B) \equiv \neg A \wedge \neg B$$

5.0.5. *Distribution of implication.*

$$A \to B \vee C \equiv (A \to C) \vee (B \to C)$$
$$A \to B \wedge C \equiv (A \to C) \wedge (B \to C)$$
$$A \to (B \to C) \equiv (A \to B) \to (A \to C)$$

5.0.6. *The Relationship Between $\leftrightarrow$ and logical equivalence.* Equivalence, $\leftrightarrow$, is a Boolean operator in propositional logic and can appear in formulas of the logic. Logical equivalence, instead, is a relations between formulas. There is potential for confusion because we are using a similar vocabulary both for the object language, in this case the language of propositional logic, and for the metalanguage that we use reason about the object language. Equivalence and logical equivalence are, nevertheless, closely related as shown by the following theorem:

Theorem 1.2. *A is logically equivalent to B if and only if $A \leftrightarrow B$ is valid*

Proof. if A is logically equivalent to B if and only if $[[A]] = [[B]]$ which is true if and only if for all $\mathcal{I}$, $\mathcal{I} \models A$ whenever $\mathcal{I} \models B$ and viceversa, and i.e., $\mathcal{I} \models A \leftrightarrow B$ for all $\mathcal{I}$, which holds iff $A \leftrightarrow B$ is valid. □

6. Truth tables

Truth tables is a simple method for explicitly enumerating all the interpretations of the propositional variables of a formula A and for each interpretation it reports the corresponding truth value of A. The truth table for a propositional formula is a table containint as many columns as the propositional variables occurring in A plus the number of subformulas of A. The truth table of A contains one row for every interpretation of prop(A), where prop(A) is the set of propositional variables that occours in A. Therefore it contains $2^{|\text{prop}(A)|}$ rows (where $|X|$ denote thor e cardinality of a set X, i.e., the number of elements that belongs to X). A raw of a truth table corresponds to an interpretation $\mathcal{I}$ i.e., an assignment of the propositional variables of A. The first n elements of the raws are the assignments of the propositional variables, while the last element is the value of $\mathcal{I}(A)$.

EXAMPLE 1.8. *the truth table of the formula $p \to (q \vee \neg r)$ is the following*[2]

p	q	r	p	$\to$	(q	$\vee$	$\neg$	r)
True	*True*	*True*	*True*	*True*	*True*	*True*	*False*	*True*
True	*True*	*False*	*True*	*True*	*True*	*True*	*True*	*False*
True	*False*	*True*	*True*	*False*	*False*	*False*	*False*	*True*
True	*False*	*False*	*True*	*True*	*False*	*True*	*True*	*False*
False	*True*	*True*	*False*	*True*	*True*	*True*	*False*	*True*
False	*True*	*False*	*False*	*True*	*True*	*True*	*True*	*False*
False	*False*	*True*	*False*	*True*	*False*	*False*	*False*	*True*
False	*False*	*False*	*False*	*True*	*False*	*True*	*True*	*False*

The truth table contains the columns corresponding to the propositinal variables of the formula $p \to (q \vee \neg r)$, i.e., $\{p, q, r\}$ and one column for every subformula of $p \to (q \vee \neg r)$, which are p, q, r, $\neg r$, $q \vee \neg r$, and $p \to (q \vee \neg r$. The truth values of the subformulas are computed starting from the simplest one to the most complex until you compute the truth value of the entire formula. (marked in red)

With the truth table for A is is possible to check if A valid, satisfiable, or unsatisfiable. If all the values in the columns of A are True, then A is valied, if there is at least one true then A is satisfiable, and all the values are False, then A is unsatisfiable. Truth tables are computationally very expensive since they enumerate the interpretations of a formula, which are esponentially large w.r.t., the size of the formula. Therefore they are only theoretical and pedagocical objects, in practice, (in real application where the number of propositional variables are large) you will never explicitly compute a truth table.

It is possible to build a truth table for more than one formula, by simply adding on the right one column for each formula. Sometimes it is also convenient to add columns corresponding to the subformulas of a complex formula. For instance if we have to compute the truth table of the formula

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

we build a truth table for all its subformulas, as follows:

F	G	$F \vee G$	$F \wedge G$	$\neg(F \wedge G)$	$(F \vee G) \wedge \neg(F \wedge G)$
True	*True*	*True*	*True*	*False*	*False*
True	*False*	*True*	*False*	*True*	*True*
False	*True*	*True*	*False*	*True*	*True*
False	*False*	*False*	*False*	*True*	*False*

Exercise 1:

Use the truth tables method to determine whether

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

is valid.

Solution

[2] To generate the truth table automatically I have used the web application availabe at `https://mrieppel.net/prog/truthtable.html`

p	q	$p \to q$	$\neg q$	$p \to \neg q$	$(p \to q) \vee (p \to \neg q)$
True	*True*	*True*	*False*	*False*	*True*
True	*False*	*False*	*True*	*True*	*True*
False	*True*	*True*	*False*	*True*	*True*
False	*False*	*True*	*True*	*True*	*True*

The formula is valid since it is satisfied by every interpretation. □

7. Propositional Theories

In the commonsense a theory is a system that allows to describe must be true in a certain domain of interests. An example of theory, which we have studied in high school, is euclidean geometry. A more sophisticate theory is the quantum mechanics. etc. Theories have been developed by humans in order to describe precisely a phenomena, and to allow to perform correct inference in order to deduce the truth of "unknown" propositions starting from a handful of principles (axioms) that are accepted to be necessarily true. In a theory there are propositions that are known to be true, i.e., somebody manage to show that truth by providing a deduction, but there are also propositions that are unknown to be true. For instance in numbner theory that are many so called "open problems" [3] In artificial intellighence, one can use a logical theory to represent the knowledge of an agent about a particular domain.by means of a logical theory, and use automatic deduction in order to infer what is true and what is false in such a domain. This was one of the original proposal of one of the founders of AI (John Mc Carthy McCarthy 1959.

A logical theory is nothing more than a set of sentences that expressews the propositions that must be true in all the configurations of the domain of interest that we believe to be possible.

Definition 1.12 (Propositional theory). *A theory is a set of propositional formulas on a set of propositional variables $\mathcal{P}$ closed under the logical consequence relation. I.e. A set of formulas T is a theory $T \models A$ implies that $A \in T$.*

An alternative and equivalent definition of theory is the following.

Definition 1.13 (Propositional theory). *A theory is a set of propositional formulas on a set of propositional variables $\mathcal{P}$ that are true in a set of interpretations of $\mathcal{P}$.*

Example 1.9. *Let $\mathcal{P}$ be the set of propositional variables. The set T of valid formulas on the propositional variables $\mathcal{P}$, i.e., $T = \{A \in \text{wffs}(\mathcal{P}) \mid A \text{ is valid}\}$. This is equivalent to say T is the set of fromulas that are true in all the interpretations of $\mathcal{P}$. For instance if p, q, r are propositional variables of $\mathcal{P}$, The formulas $p \vee \neg p$, $q \vee \neg q$, $p \wedge q \to p$, $r \to r$ belongs to T. While the formulas $p \to q$ does not. Notice that T is closed under logical consequence. Indeed suppose that $A_1, \dots, A_n \models A$, and $A_1, \dots, A_n \in T$. Then we have that each A_i is valid, and therefore it is true in every models, The fact that $A_1, \dots, A_n \models A$ entails that A is also treue in every interpretation, and therefore A is valied, hwnce it belongs to T.*

[3] An example of open problem in number theory is connected to the Erdös–Moser equation:

$$1^k + 2^k + \cdots + m^k = (m+1)^k,$$

where m and k are positive integers. The only known solution is $1^1 + 2^1 = 3^1$, and Paul Erd ös conjectured that no further solutions exist.

Example 1.10. *is the set of formulas which are true in the the following three interpretations of* $\mathcal{P}$*,* $\{\mathcal{I}_1, \mathcal{I}_2, \mathcal{I}_3\}$*, where* $\mathcal{I}_1 = \{p, q\}$*,* $\mathcal{I}_2 = \{q, r\}$ *and* $\mathcal{I}_2 = \{p, r\}$*. Notice that this theory contains all the valid formulas since they are true in all the interpretations, and therefore also in* $\mathcal{I}_1, \mathcal{I}_2$ *and* $\mathcal{I}_3$*. However* T *contains formulas which are not valid as for instance* $p \vee q$*,* $\neg p \rightarrow q \wedge r$*. Notice that* T *can be defined as the set of formulas that are logical consequences of the formula* $A = (p \wedge q) \vee (p \wedge r) \vee (q \wedge r)$*. Indeed if* B *is a logical consequence of* A*, it is true in all the models that satisfies* A*. Since* $\mathcal{I}_1, \mathcal{I}_2$ *and* $\mathcal{I}_3$ *satisfies* B*, then* $B \in T$*. We way that* A *is an* axiomatization *of* T*, because from the axiom* A *all the formulas of* T *follows logically.*

Example 1.11. *Let* T *be the set of formulas that are true in a single interpretation* $\mathcal{I}$*.* T *is a theory since it is closed under logical consequence. Indeed if* $A_i \in T$ *we have that* $\mathcal{I} \models A_i$ *we have that* $\mathcal{I} \models A_i$*, and if* $A_1, \ldots, A_n \models A$*, we have that* $\mathcal{I} \models A$ *and therefore* $A \in T$*. This theory is* complete *in the sense that for every formula* A*, either* $A \in T$ *or* $\neg A \in T$*.*

The three examples of theories provided above, range from the weakest theory to the strongest one.

A propositional theory contains an infinite set of formulas. Indeed every theory contains all the propositional tautoloties, which are infinite. However thee infinite set of formulas could be defined as the logical consequences of a smaller set of formulas (possibily but not necessarily finite) such that all the formulas of the theory logically follows. These set of formulas are called *axioms* of a theory, They are the basic principles of the theory from which everything else that is true in the theory follows logically. So the set of axioms constitute a base for the theory and characterizes it.

Definition 1.14. *A set* S *of formulas is a set of* axioms *(or equivalently an* axiomatization*) of a theory* T *if* $T = \{A \in \text{wffs}(\mathcal{P}) \mid S \models A\}$*.*

An important property for a set of axioms S of a theory T is that they are minimal, in the sense that no formulas in S is a logical consequences of other formulas in S. This property can be formulated also as follows: there is no $S' \subset S$ that is an axiomatization of T.

Example 1.12. *The axiomatization of the theory of Example 1.9 is the empty set. An axiomatization of the theory of Example 1.11 is the set* $\{p \mid \mathcal{I} \models p\} \cup \{\neg p \mid \mathcal{I} \not\models p\}$

8. The Compactness Theorem

In this section we prove a fundamental result about propositional logic called the *Compactness Theorem*. This will play an important role in the second half of the course when we study predicate logic. This is due to our use of Herbrand's Theorem to reduce reasoning about formulas of predicate logic to reasoning about infinite sets of formulas of propositional logic. Before stating and proving the Compactness Theorem we need to introduce one new piece of terminology.

Recall that a set of formulas Γ is satisfiable if there is an assignment that satisfies every formula in Γ. For example, the set of formulas

$$\Gamma = \{p_1 \vee p_2, \neg p_2 \vee \neg p_3, p_3 \vee p_4, \neg p_4 \vee p_5, \ldots\}$$

on the infinite set of propositional variables $\mathcal{P} = \{p_1, p_2, p_2, \dots\}$ is satisfied by the assignment $\mathcal{I}$ defined as

$$\mathcal{I}(p_i) = \begin{cases} \text{True} & \text{if } i \text{ is odd} \\ \text{False} & \text{if } i \text{ is even} \end{cases}$$

DEFINITION 1.15. *A set of formula is* finitely satisfiable *if all its finite subsets are satisfiable.*

THEOREM 1.3 (Compactness Theorem). *A set of formulas Γ is satisfiable if and only if it is finitely satisfiable.*

Notice that the formulation above of the compactness theorem is is different fron the one introduced by Theorem 1.1. Nevertheless, if we proove Theorem 1.3, we can easily prove the original compactness theorem by combining the refutation principle, monotonicity and the the new formulaiton of the compactness theorem.

- $\Gamma \models A$ if and only if $\Gamma \cup \{\neg A\}$ is not satisfiable (by the refutation principle)
- $\Gamma \cup \{\neg A\}$ is not satisfiable if and only if there is a finite subset Γ_0 of $\Gamma \cup \{\neg A\}$ which is not satisfiable (new formulation of the compactness theorem).
- This implies that $\Gamma_0 \cup \{\neg A\}$ is not satisfiable, by monotonicity, and therefore by refutation principle that $\Gamma_0 \models A$.

To get an idea of what says the Compactness theorem consider the following intuitive example

EXAMPLE 1.13. *Suppose that you have a logical language in which you can express by means of formula K, P_i and S the following proposition:*

K	*there is a cake of finite size*
P_i	*The i-th person has a piece of cake for $i = 1, 2, 3, \dots s$*
S	*The pieces of cake have all the same non zero dimension*

The formula $K \wedge S \to P_i$ formalizes the fact that if there is a cake and it is equally divided then the ith person gets it's piece of cake. Notice consider any finite subset of the set $\Gamma = \{K \wedge S \to P_1, K \wedge S \to P_2, K \wedge S \to P_3, \dots\}$. i.e., for every finite set of natural numbers $I \subset \mathbb{N}$ let

$$\Gamma_I = \{K \wedge S \to P_i \mid i \in I\}$$

Notice that the whole Γ is not satisfiable, since you cannot cut a finite cake in an infinite set of slices of finite and constant size. However each Γ_I for every finite $I \subset \mathbb{N}$ is satisfiable. Since the compactness theorems holds in propositional logic, we can conclude that such a scenario cannot be formalized in propositional logic.

To prove the compactness theorems we first need to prove the following lemma:

LEMMA 1.1. *If Γ is finitely satisfiable then either $\Gamma \cup \{\phi\}$ or $\Gamma \cup \{\neg\phi\}$ is finitely satisfiable, for every formula ϕ.*

PROOF.
- Suppose the conclusion of the lemma does not hold: Both $\Gamma \cup \{\phi\}$ and $\Gamma \cup \{\neg\phi\}$ are not finitely satisfiable.
- Hence, there are two finite subsets Γ_1 and Γ_2 of Γ such that both $\Gamma_1 \cup \{\phi\}$ and $\Gamma_2 \cup \{\neg\phi\}$ are not satisfiable.
- Let us show that $\Gamma_1 \cup \Gamma_2$ does not have models, i.e., it is unsatisfiable.

- If $\mathcal{I}$ is a model of Γ_1, than it cannot be a model of ϕ, therefore it is a model of $\neg\phi$. Otherwise $\Gamma_1 \cup \{\phi\}$ would be satisfiable. Therefore, $\mathcal{I} \models \neg\phi$.
- Since $\Gamma_2 \cup \{\neg\phi\}$ is not satisfiable, then $\mathcal{I}$ cannot be a model of Γ_2.
- This implies that every model of Γ_1 is not a model of Γ_2 and therefore, $\Gamma_1 \cup \Gamma_2$ is not satisfiable.
- Since $\Gamma_1 \cup \Gamma_2$ is finite and it is a subset of Γ, then Γ cannot be finitely satisfiable. This contraddicts the hypothesis of the lemma, and therefore the lemma is proved.

□

Let us now prove the compactness theorem:

Of theorem 1.3. If Γ is satisfiable, then every subset of Γ is satisfiable, and therefore Γ is finitely satisfiable. The prove of the opposite direction is more complex. We have to show that if Γ is finitely satisfiable then the whole Γ is satisfiable, i.e., there is an interpretation $\mathcal{I}$ such that $\mathcal{I} \models \Gamma$.

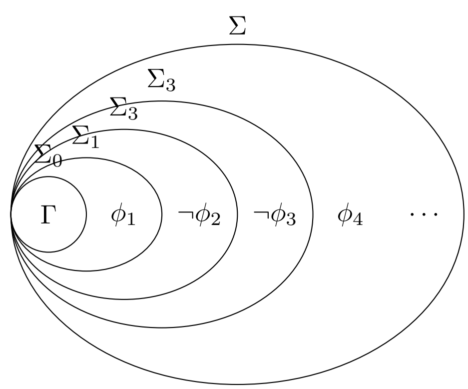
- let enumerate all the formulas $\phi_1, \phi_2, \phi_3, \dots$ of the language of Γ.
- let us define the sequence $\Sigma_0, \Sigma_1, \Sigma_2, \dots$ as follows

$$\Sigma_0 = \Gamma \qquad \Sigma_n = \begin{cases} \Sigma \cup \{\phi_n\} & \text{if } \Sigma_{n-1} \cup \{\phi_n\} \text{ is fin. sat.} \\ \Sigma \cup \{\neg\phi_n\} & \text{if } \Sigma_{n-1} \cup \{\neg\phi_n\} \text{ is fin. sat.} \end{cases}$$

- By induction, using previous lemma, Σ_i is finitely satisfiable;
- Let

$$\Sigma = \bigcup_{n \geq 0} \Sigma_i$$

The construction of Σ is shown in the following picture:



- By construction Σ is finitely satisfiable. Furthermore
  (1) For every formula ϕ either $\phi \in \Sigma$ or $\neg\phi \in \Sigma$ but not both.
  (2) For every $p \in \mathcal{P}$, $p \in \Sigma$ or $\neg p \in \Sigma$ but not both.
  (3) We define the interpretation $\mathcal{I}(p) = \begin{cases} \text{True} & \text{if } p \in \Sigma \\ \text{False} & \text{if } \neg p \in \Sigma \end{cases}$
  (4) Let us show that $\mathcal{I} \models \phi$ for all $\phi \in \Sigma$.
  (5) Consider the finite set Σ_i that contains ϕ and either p or $\neg p$ for all propositional variable p that occours in ϕ. Since ϕ contains only a finite set of propositional variables, such an finite Σ_i exists.

(6) Since Σ_i is finite, and Σ is finitely satisfiable, there is an interpretation $\mathcal{I}'$ that satisfies Σ_i, and therefore $\mathcal{I}' \models \phi$
(7) Furthermore $\mathcal{I}' \models p$ of $p \in \Sigma_i$ or $\mathcal{I}' \models \neg p$ if $\neg p$ in Σ_i.
(8) However by construction of $\mathcal{I}$, we have that $\mathcal{I}'$ and $\mathcal{I}$ agree on all the interpretations of all the propositional variables of ϕ and therefore $\mathcal{I} \models \phi$.
(9) Hence, $\mathcal{I} \models \Sigma$.
(10) Since $\Gamma \subset \Sigma$, then $\mathcal{I} \models \Gamma$.

□

9. Exercises

9.1. Formulas, subformulas, and other syntactic objects.

Exercise 2:
Decide which of the following phrases expresses a proposition

(1) The dog of my syster
(2) Is mario Happy?
(3) A lion in the forest
(4) A tiger is walking in the forest
(5) the sooner the better
(6) No more food please!
(7) Have a nice week end!
(8) The hause in which I was born
(9) Closed door
(10) The door is closed

Exercise 3:
Knowing that the precedence relations batween the propositional connectives is

$$\text{“}\neg\text{”} \prec \text{“}\wedge\text{”} \prec \text{“}\vee\text{”} \prec \text{“}\rightarrow\text{”} \prec \text{“}\leftrightarrow\text{”}$$

add the parenthesis to specify the correct parsing of the following formulas:

(1) $(\neg p \vee q) \wedge (q \rightarrow (\neg r \wedge \neg p)) \wedge (p \vee r)$
(2) $((\neg p \vee q) \rightarrow q \wedge (q \rightarrow r) \wedge \neg r) \rightarrow p$
(3) $\neg p \wedge (\neg q \vee r) \wedge (\neg p \rightarrow q \wedge \neg r)$

Solution

$$(\neg p \vee q) \wedge (q \rightarrow (\neg r \wedge \neg p)) \wedge (p \vee r)$$
$$((\neg p) \vee q) \wedge (q \rightarrow ((\neg r) \wedge (\neg p))) \wedge (p \vee r)$$

$$((\neg p \vee q) \rightarrow q \wedge (q \rightarrow r) \wedge \neg r) \rightarrow p$$
$$(((\neg p) \vee q) \rightarrow (q \wedge (q \rightarrow r) \wedge (\neg r))) \rightarrow p$$

$$\neg p \wedge (\neg q \vee r) \wedge (\neg p \rightarrow q \wedge \neg r)$$
$$(\neg p) \wedge ((\neg q) \vee r) \wedge ((\neg p) \rightarrow (q \wedge (\neg r)))$$

□

Exercise 4:
When we have two connectives which are the same, then we give precedence to the right one. I.e., $a \circ b \circ c$ reads as $a \circ (b \circ c)$ for every binary connective $\circ \in \{\rightarrow, \wedge, \vee, \leftrightarrow\}$. Put the right parentesist on the following formulas

(1) $a \rightarrow b \wedge \neg c \rightarrow d$

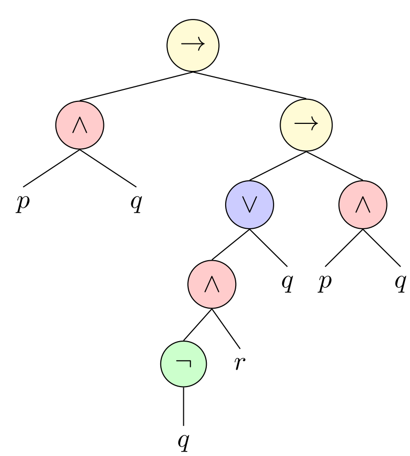
(2) $a \to b \to c \to d$
(3) $a \leftrightarrow b \leftrightarrow c \wedge d \leftrightarrow e$

Exercise 5:
Draw the formula tree of the following formula:

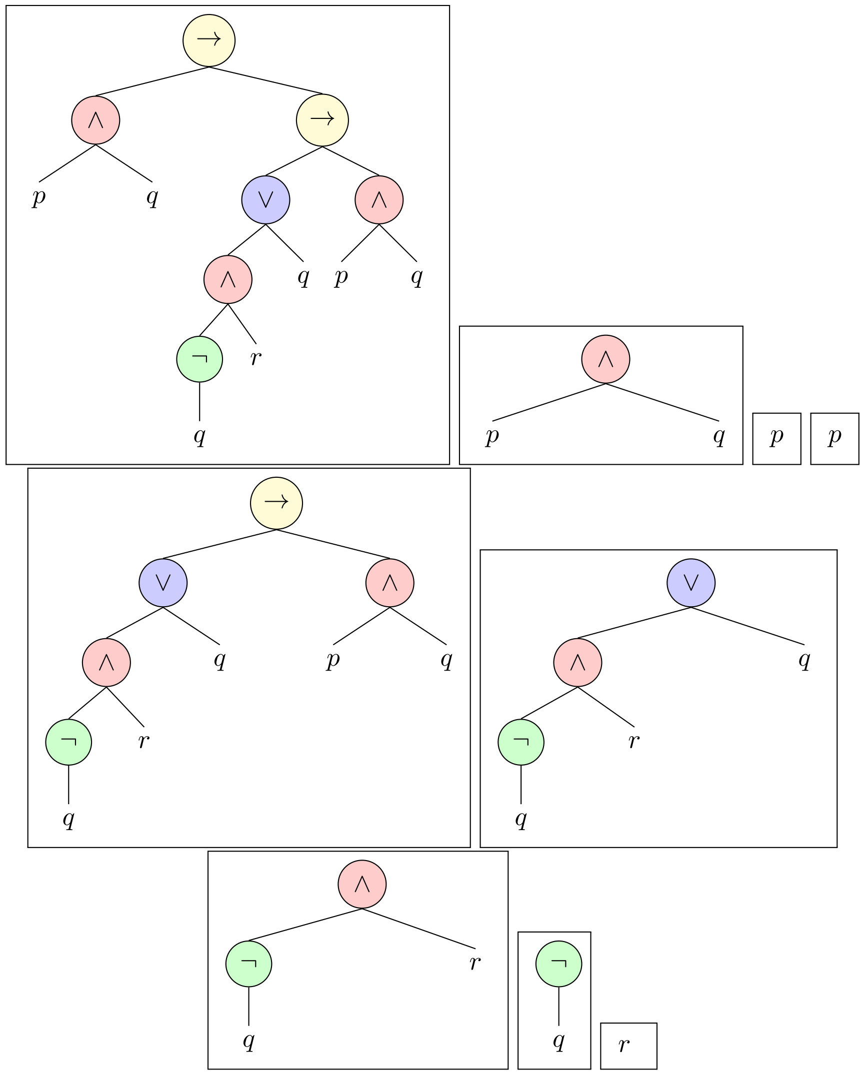
$$(p \wedge q) \to \neg q \wedge r \vee q \to (p \wedge q)$$

Rimember that the expression $a \to b \to c$ is parsed as $a \to (b \to c)$, and $a \leftrightarrow b \leftrightarrow c$ is parsed as $a \to (b \to c)$. Count all the sub-formulas of ϕ.

Solution The formula ϕ is parsed as $p \wedge q \to (((\neg q \wedge r) \vee q) \to q \wedge p)$ where parentesis are made explicit. The tree of ϕ is



In general a formula has as many subformulas as many distinct subtrees of its formula tree, including the entire tree and the leavers. Therefore ϕ has 9 subformula corresponding to the following subtrees:



□

Exercise 6:

Draw the formula tree for the following formulas and count the number of subformulas:

(1) $a \to b \to c \leftrightarrow a \wedge b \vee \neg a \wedge \neg b \vee \neg a$
(2) $\neg(a \leftrightarrow b \leftrightarrow c)$
(3) $\neg(a \wedge \neg(b \wedge \neg c)))$.
(4) $a \wedge b \vee b \wedge a$

Exercise 7:

For each of the following formula draw the formula tree and the form with minimal number of parenthesis:

(1) $(p \wedge q) \to (\neg(q \to r \wedge \neg r))$
(2) $((p \leftrightarrow (\neg\neg q)) \vee r \wedge q$

Exercise 8:

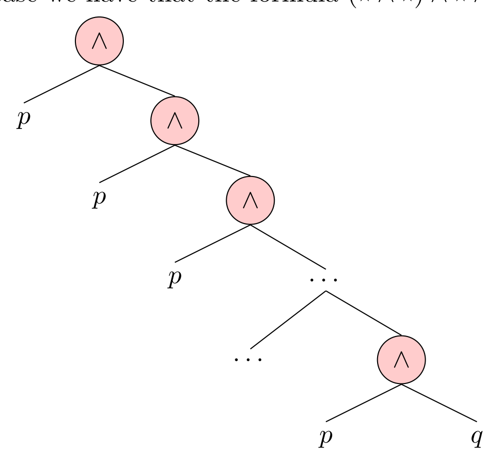
Produce a formula that contains only 2 propositional variables and $n = 1, 2, 3, 4$ binary connective $\wedge$, which has the maximum number of subformulas. Can you generalize this process to any n?

Solution If $n = 1$ then the formula $p \wedge q$ has 3 subformulas, and this is the only formula that satisfies the requirements of containing two propositinal variables and one binary connective.

With $n = 2$, then the formula must be of the form $\star \wedge (\star \wedge \star)$ or $(\star \wedge \star) \wedge \star$ where $\star$ can be replaced either with p or q. This implies that the atomic subformulas must be at most 2. Since the fromula contains 2 coonectives and every coonective corresponds to a subformula, then we have that the formulas has 4 subformulas.

if $n = 3$ the possible shapes is $\star \wedge (\star \wedge (\star \wedge \star))$, where $\star$ can be replaced by p or q. Furthermore notice that all the subformulas that has $\wedge$ as main connective are different. From this we can conclude that the we can produce a formula with 3 connectives that has 3 subformulas + 2 atomic formulas which in total is equal to 5. Notice that we can construct formulas that contains 3 occurrences of $\wedge$ that has less subformulas. For instance $(p \wedge q) \wedge (p \wedge q)$, contains only 4 subformulas, since the subformula $p \wedge q$ occurs twice and therefore it contributes only for 1 to the total number of subformulas.

In the general case we have that the formula $(\star \wedge \star) \wedge \star \wedge \star$



contains $n + 2$ subformulas. $\square$

Exercise 9:

Suppose that a formula ϕ contains n occurrences of $\wedge$, m occurrences of $\vee$ and p occurrences of $\neg$.

(1) What is the maximum number s of subformulas of ϕ?
(2) Explain how you get this result.
(3) Provide an example of a formula with s subformulas;
(4) Provide an example of a formula with less then s subformulas. maximum number.

Solution

(1) For every occurrence of $\vee$ and $\wedge$ the formula has two subformulas (they can be different provided that we have anought propositional variables),

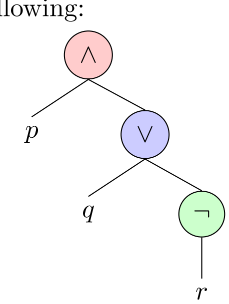
for every occurrence of $\neg$ there is one subformula. In total therefore we have $s = 2^{n+m} + p + 1$

(2) If you imagine the formula tree, and every node labelled with n there two subtrees, and for every nodel labelled with $\neg$ there is only one subtree. In both case we have to consider also the tree as a formula is a subformula of itself. If we make sure that all the subtrees are different, by using diffferent set of propositinal variables for each subtree, we can guarantee that all the subtrees are different.

(3) For instance $p \wedge (q \vee negr)$ is a formula that contains 1 $\wedge$, 1 occurrence of $\vee$ and 1 occurrence of $\neg$ and has $2^{1+1} + 1 + 1 = 6$ subformulas, which are

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

The formula tree is the following:



which has exactly 6 nodes.

(4) To obtain less subformulas than s we have to make sure that two subformulas are the same. This is the case for instance in the simple formula

$$p \wedge p$$

that contains 1 conjunction connective. Therefore $s = 2^1 + 1 = 3$, but the number of subformulas are 2, i.e., $p \wedge p$ and p. This is due to the fact that p is a subformula of both branches of $\wedge$.

□

Exercise 10:

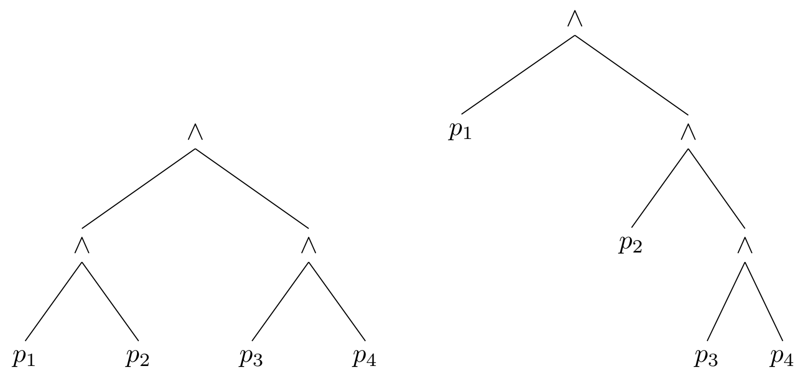
If a formula ϕ contains n connectives, what is the minimum and the maximum number of subformulas?

Solution First of all notice that the connective $\neg$ applies only to one subformula, while all the other connectives are binary, i.e., they apply to two subformulas. To minimize the number of subformulas therefore we can suppose that all the n connectives are $\neg$, i.e., the formula is of the form:

$$\overbrace{\neg(\neg(\neg(\dots(\neg}^{\times n} p)\dots))$$

In this case the formula has $n + 1$ subformulas. Instead to maximize the number of subformulas one should use binary connectives since for each binary connective we can potentially have 2 subformulas. The maximum number of subformulas using n connectives can be obtained by using only binary connectives and making sure that the two subformulas of all the binary connectives are distinct. This can be obtained by introducing enough propositional variables to make all subformulas for every connective different from the other. This implies that the occurrence of every connective corresponds to two subformulas. Therefore the maximum number of subformulas are $2n+1$

Suppose for instance that $k = 2$, then the following are two examples of formulas with binary connectives ($\wedge$ in this case):



The number of subformuals are $2 \cdot 3 + 1 = 7$. More in general the maximum number of subformulas are $2n - 1$. □

Exercise 11:

Prove by structural induction that a propositional formula that contains n occurrences of $\neg$ and m occurrences of binary connectives has at most $n + 2m + 1$ subformulas. **Solution** Let ϕ be a formula that contains n occurrences of $\neg$

and m occurrences of binary propositional connectives. We prove the theorem by induction on $n + m$

Base case: Suppose that $n + m = 0$, then ϕ does not contain any propositoinal connective, and therefore it is an atomic formula p, which has exactly 1 subformula, i.e., ϕ itself. Therefore the number of subformulas of ϕ are $1 = n + 2m + 1$.

Step case: Suppose that the property holds for all the formulas that contains m' occurrences of $\neg$ and n' occurrences of binary connectives with $m' \leq m$ and $n' \leq n$ and $m' + n' < m + n$, and let us prove that it holds also for $m + n$. We consider two cases.

(1) ϕ is of the form $\neg\phi_1$. Then ϕ_1 contains $n-1$ occurrences of $\neg$ and m occurrences of binary connectives. By induction hypothesis we can infer that ϕ_1 has at most $n-1+2m+1 = n+2m$ subformulas. Since the subformulas of $\neg\phi_1$ are the subformulas of ϕ_1 plus the formula $\neg\phi_1$ itself, then the maximum number of subformulas of $\neg\phi_1$ are $n + 2m + 1$.

(2) ϕ is of the form $\phi_1 \circ \psi_2$ for some binary connective $\circ \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$. Then we have that each ϕ_i with $i = 1, 2$ contains n_i occurrences of $\neg$ and m_i occurrences of binary connectives. with $m_i \leq m - 1$, and $n_i \leq n$. We therefore have that $n_i + m_i < n + m$. By induction on ϕ_i we have that the maximum number of subformulas of ϕ_i are $n_i + 2m_i + 1$. Since the subformulas of $\phi_1 \circ \phi_2$ are the subformulas of ϕ_i for $i = 1, 2$ plus $\phi_1 \circ \phi_1$ itself, we have that $\phi_1 \circ \phi_2$ has at most

$n_1 + 2m_1 + 1 + n_2 + 2m_2 + 1 + 1 = (n_1 + n_2) + 2(m_1 + m_2 + 1) + 1$ subformulas. Notice that $n_1 + n_2 = n$ since every occurrence of $\neg$ in ϕ is either an occurrence in ϕ_1 or in ϕ_2 but not in both. Furthermore $m_1 + m_2 = m - 1$ because an occurrence of a binary connective is either in ϕ_1 or in ϕ_2 or it is $\circ$, the main connective of $\phi_1 \circ \phi_2$. This allows us to infer that $\phi_1 \circ \phi_2$ has at most $n + 2m + 1$ subformulas.

□

9.2. Semantics of propositional logic. Exercise 12:

Define when a formula is valid, satisfiable, unsatisfiable, and not valid, and the relations between these concepts.

Exercise 13:

Draw an arrow between the statement of the left colong and the one of the right column every time the former implies the latter. Then repeat the exercize swapping the columns.

(a) A is satisfiable	(1) $\neg A$ is satisfiable
(b) B is satisfiable	(2) $\neg B$ is satisfiable
(c) A is satisfiable and B is satisfiable	(3) $\neg A$ is valid
(d) A is satisfiable or B is satisfiable	(4) $\neg B$ is valid
(e) A is valid	(5) $\neg A$ is unsatisfiable
(f) B is valid	(6) $\neg B$ is unsatisfiable
(g) A is valid and B is valid	(7) $\neg A$ is not valid
(h) A is valid or B is valid	(8) $\neg B$ is bot valid
(i) A is unsatisfiable	(9) $A \wedge B$ is satisfiable
(j) B is unsatisfiable	(10) $A \wedge B$ is valid
(k) A is unsatisfiable and B is unsatisfiable	(11) $A \wedge B$ is unsatisfiable
(l) A is unsatisfiable or B is unsatisfiable	(12) $A \wedge B$ is not valid
(m) A is not valid	(13) $A \vee B$ is satisfiable
(n) B is not valid	(14) $A \vee B$ is valid
(o) A is not valid and B is not valid	(15) $A \vee B$ is unsatisfiable
(p) A is not valid or B is bot valid	(16) $A \vee B$ is not valid

Exercise 14:

Explain the difference between the following statements

(1) $\models A \vee B$ ($A \vee B$ is valid)
(2) $\models A$ or $\models B$ (A is valid or B is valid)

which one is the strongest?

Solution Let us expland 1. and 2. according to the definition of valid formula:

(1) $\models A \vee B$ means that for every interpretation $\mathcal{I}$, $\mathcal{I} \models A \vee B$, which means that either $\mathcal{I} \models A$ or $\mathcal{I} \models B$
(2) $\models A$ or $\models B$ instead means that either for every interpretation $\mathcal{I}$, $\mathcal{I} \models A$ or for every interpretation $\mathcal{I}$, $\mathcal{I} \models B$.

To highlight the difference between 1. and 2. you can write their definition by using a more formal notation,

$$\models A \vee B \iff \forall \mathcal{I}, (\mathcal{I} \models A \text{ or } \mathcal{I} \models B) \tag{5}$$

$$\models A \text{ or } \models B \iff (\forall \mathcal{I}, \mathcal{I} \models A) \text{ or } (\forall \mathcal{I}, \mathcal{I} \models B) \tag{6}$$

An example that shows the difference can be constructed by taking A equal to the atomic formula p and B the negated atomic formula $\neg p$. You have that $\models p \vee \neg p$, but neither $\models p$ nor $\models \neg p$ Clearly 2. is a stronger statement than 1. $\square$

Exercise 15:

Find three formula A, B, and C such that $A \wedge B \wedge C$ is unsatisfiable and such that the conjunction of any pair of them is satisfiable. I.e., $A \wedge B$, $A \wedge C$ and $B \wedge C$ are satisfiable.

Exercise 16:

Suppose that A and B contains two disjoint set of propositional variables. Show that $A \wedge B$ is satisfiable if and only if A is satisfiable and B is satisfiable.

Exercise 17:

Show that if A is satisfiable then $A \wedge p$ or $A \wedge \neg p$ is satisfiable.

Exercise 18:

Prove that if A and B contain a disjoint set of propositional variable then $A \vee B$ is valid if and only if A is valid or B is valid.

Exercise 19:

Let ϕ be a propositional formula built only with the operators $\vee$, $\wedge$ and $\neg$. An occurrence of a propositional variable p in ϕ is *positive* if it is in the scope of an *even* number of $\neg$ operators, and it is negative if it is not positive. Provide an explanation (or better a proof) of the fact that if ϕ does not contains two occurrences of the same propositional variable, which are one positive and one negative, then ϕ is satisfiable.

Solution If you transform ϕ in NNF (negated normal form) then for every propositional variable p, either all its occurrences in NNF(ϕ) will not be negated (if all the occurrences of p in ϕ are positive, then an even number negations cancel out) or all of them will have a $\neg$ in front of them (i.e, when they are negative occurrences, an odd number of negations reduce to a single negation). The assignment that maps all positive p into true and the negative p into false, satisfies NNF(ϕ) and therefore it satisfies also ϕ. $\square$

9.3. Truth tables. Exercise 20:

For each of the following formulas, construct a truth table and state whether it is valid, satisfiable, or unsatisfiable.

(1) $p \wedge \neg p$
(2) $p \vee \neg p$
(3) $(p \vee \neg q) \rightarrow q$
(4) $(p \vee q) \rightarrow (p \wedge q)$
(5) $(p \rightarrow q) \leftrightarrow (\neg q \rightarrow \neg p)$
(6) $(p \rightarrow q) \leftrightarrow (q \rightarrow p)$

Exercise 21:

Give a truth-table definition of the ternary boolean operation "if p then q else r", and write a propositional formula using only the connectives $\rightarrow$ and $\neg$ that is equivalent to such an operator.

Solution One possible intuitive reading of "if p then q else r" is that when p is true then q should be also true and we don't know anything about r, and when p is false then q should be true, and we don't know anything about p. This is represented by the following table:

p	q	r	if p then q else r
T	T	T	T
T	T	F	T
T	F	T	F
T	F	F	F
F	T	T	T
F	T	F	F
F	F	T	T
F	F	F	F

Such a ternary connective can be formalized by the propositional formula

$$(p \rightarrow q) \wedge (\neg p \rightarrow r)$$

□

Exercise 22:

Given the truth table for an arbitrary n-ary boolean function

$$f : \{0,1\}^n \rightarrow \{0,1\}$$

describe how one can build a formula ϕ using only n propositional variables $p_1, \ldots, p_n$ such that the following holds:

$$f(x_1, \ldots, x_n) = 1 \text{ if and only if } \mathcal{I} \models \phi$$

where $\mathcal{I}$ is an assignment such that $\mathcal{I}(p_i) = x_i$

Solution For every $\boldsymbol{x} = (x_1, \ldots, x_n) \in \{0,1\}^n$ let us define the formula $\phi_{\boldsymbol{x}}$ as follows:

$$\phi_{\boldsymbol{x}} = \bigwedge_{\substack{i=1 \\ x_i=1}}^{n} p_i \wedge \bigwedge_{\substack{i=1 \\ x_i=0}}^{n} \neg p_i$$

Notice that the formula $\phi_{\boldsymbol{x}}$ is satisfied only by a single assignment, i.e., the assignment in which $\mathcal{I}(p_i) = x_i$ for all the propositional variables p_i. Let us define ϕ as the disjunction of all the $\phi_{\boldsymbol{x}}$ such that $f(\boldsymbol{x}) = 1$.

$$\phi = \bigvee_{\substack{\boldsymbol{x} \in \{0,1\}^n \\ f(\boldsymbol{x})=1}} \phi_{\boldsymbol{x}}$$

For every interpretation $\mathcal{I}$, if $\mathcal{I} \models \phi$ then for some $\boldsymbol{x}$ for which $f(\boldsymbol{x}) = 1$, $\mathcal{I} \models \phi_{\boldsymbol{x}}$. By the construciton of $\phi_{\boldsymbol{x}}$ $\mathcal{I}$ is the assignment that assigns x_i to each p_i. □

Exercise 23:

Are the following formulae satisfiable, valid, unsatisfiable, or not valid?

(1) $(\neg p \vee q) \wedge (q \rightarrow (\neg r \wedge \neg p)) \wedge (p \vee r)$
(2) $((\neg p \vee q) \rightarrow q \wedge (q \rightarrow r) \wedge \neg r) \rightarrow p$
(3) $\neg p \wedge (\neg q \vee r) \wedge (\neg p \rightarrow q \wedge \neg r)$

Solution Let us build the truth tables of the three formulas:

p	q	r	$(((\neg$	$(p$	$\vee$	$q))$	$\wedge$	$(q$	$\rightarrow$	$((\neg$	$r)$	$\wedge$	$(\neg$	$p))))$	$\wedge$	$(p$	$\vee$	$r))$
T	T	T	F	T	T	T	F	T	F	F	T	F	F	T	**F**	T	T	T
T	T	F	F	T	T	T	F	T	F	T	F	F	F	T	**F**	T	T	F
T	F	T	F	T	T	F	F	F	T	F	T	F	F	T	**F**	T	T	T
T	F	F	F	T	T	F	F	F	T	T	F	F	F	T	**F**	T	T	F
F	T	T	F	F	T	T	F	T	F	F	T	F	T	F	**F**	F	T	T
F	T	F	F	F	T	T	F	T	T	T	F	T	T	F	**F**	F	F	F
F	F	T	T	F	F	F	T	F	T	F	T	F	T	F	**T**	F	T	T
F	F	F	T	F	F	F	T	F	T	T	F	T	T	F	**F**	F	F	F

p	q	r	$((((\neg$	$p)$	$\vee$	$q)$	$\rightarrow$	$((q$	$\wedge$	$(q$	$\rightarrow$	$r))$	$\wedge$	$(\neg$	$r)))$	$\rightarrow$	$p)$
T	T	T	F	T	T	T	F	T	T	T	T	T	F	F	T	**T**	T
T	T	F	F	T	T	T	F	T	F	T	F	F	F	T	F	**T**	T
T	F	T	F	T	F	F	T	F	F	F	T	T	F	F	T	**T**	T
T	F	F	F	T	F	F	T	F	F	F	T	F	F	T	F	**T**	T
F	T	T	T	F	T	T	F	T	T	T	T	T	F	F	T	**T**	F
F	T	F	T	F	T	T	F	T	F	T	F	F	F	T	F	**T**	F
F	F	T	T	F	T	F	F	F	F	F	T	T	F	F	T	**T**	F
F	F	F	T	F	T	F	F	F	F	F	T	F	F	T	F	**T**	F

p	q	r	$(((\neg$	$p)$	$\wedge$	$((\neg$	$q)$	$\vee$	$r))$	$\wedge$	$((\neg$	$p)$	$\rightarrow$	$(q$	$\vee$	$(\neg$	$r))))$
T	T	T	F	T	F	F	T	T	T	**F**	F	T	T	T	T	F	T
T	T	F	F	T	F	F	T	F	F	**F**	F	T	T	T	T	T	F
T	F	T	F	T	F	T	F	T	T	**F**	F	T	T	F	F	F	T
T	F	F	F	T	F	T	F	T	F	**F**	F	T	T	F	T	T	F
F	T	T	T	F	T	F	T	T	T	**T**	T	F	T	T	T	F	T
F	T	F	T	F	F	F	T	F	F	**F**	T	F	T	T	T	T	F
F	F	T	T	F	T	T	F	T	T	**F**	T	F	F	F	F	F	T
F	F	F	T	F	T	T	F	T	F	**T**	T	F	T	F	T	T	F

(1) The first formula is not valid since there are truth assignments that evaluates it to false. It is satisfiable since for the truth assignment $\mathcal{I}(p) = False$, $\mathcal{I}(q) = False$ and $\mathcal{I}(r) = True$ (one but last line of the first truth table) the formula is evaluated to True. Consequenclty the formula is not unsatisfiable.
(2) The second formula is valid since it is true for all the assignments. It is also satisfiable since there are assignments that makes it true.
(3) The third formula, like the first one is not valid but it is satisfiable.

□

Exercise 24:

Suppose that ϕ contains only the $\leftrightarrow$ operator. Prove that if every propositional variable p occours an evenq number of times, then ϕ is valid. **Solution** The proof

is based on the fact that $\leftrightarrow$ operator is associative and commutative. let us prove these two properties

Associativity of $\leftrightarrow$: We build the truth tables of $A \leftrightarrow (B \leftrightarrow C)$ and $(A \leftrightarrow B) \leftrightarrow C$ and see that the two formulas takes identical truth values.

A	B	C	A	$\leftrightarrow$	(B	$\leftrightarrow$	C)	(A	$\leftrightarrow$	B)	$\leftrightarrow$	C
T	T	T	T	T	T	T	T	T	T	T	T	T
T	T	F	T	F	T	F	F	T	T	T	F	F
T	F	T	T	F	F	F	T	T	F	F	F	T
T	F	F	T	T	F	T	F	T	F	F	T	F
F	T	T	F	F	T	T	T	F	F	T	F	T
F	T	F	F	T	T	F	F	F	F	T	T	F
F	F	T	F	T	F	F	T	F	T	F	T	T
F	F	F	F	F	F	T	F	F	T	F	F	F

Commutativity of $\leftrightarrow$: We use the same procedure to prove that $A \leftrightarrow B$ is equivalent to $B \leftrightarrow A$.

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

Associativity and commutativity imply that every formula ϕ in which all the propositional variables appears an even number of ties can be rearranged in the following form

$$(p_1 \leftrightarrow p_1) \leftrightarrow (p_2 \leftrightarrow p_2) \leftrightarrow (p_3 \leftrightarrow p_3), \dots$$

Which is equivalent to $\top \leftrightarrow \top \leftrightarrow \top \dots$ which is always true. □

9.4. Logical consequence. Exercise 25:

Prove the following logical consequences:

(1) $p \models p \vee q$
(2) $q \vee p \models p \vee q$
(3) $p \vee q, p \rightarrow r, q \rightarrow r \models r$
(4) $p \rightarrow q, p \models q$
(5) $p, \neg p \models q$

Solution

(1) Suppose that $\mathcal{I} \models p$, then by definition $\mathcal{I} \models p \vee q$.
(2) Suppose that $\mathcal{I} \models q \vee p$, then either $\mathcal{I} \models q$ or $\mathcal{I} \models p$. In both cases we have that $\mathcal{I} \models p \vee q$.
(3) Suppose that $\mathcal{I} \models p \vee q$ and $\mathcal{I} \models p \rightarrow r$ and $\mathcal{I} \models q \rightarrow r$. Then either $\mathcal{I} \models p$ or $\mathcal{I} \models q$. In the first case, since $\mathcal{I} \models p \rightarrow r$, then $\mathcal{I} \models r$, In the second case, since $\mathcal{I} \models q \rightarrow r$, then $\mathcal{I} \models r$.
(4) Suppose that $\mathcal{I} \models \neg p$, then not $\mathcal{I} \models p$, which implies that there is no $\mathcal{I}$ such that $\mathcal{I} \models p$ and $\mathcal{I} \models \neg p$. This implies that all the interpretations that satisfy p and $\neg p$ (actually none) satisfy also q.
(5) ...
(6) ...

□

Exercise 26:

Show that if $\Gamma \models A$ and $\Gamma \models \neg A$, then Γ is not satisfiable.

Exercise 27:
Prove that $q \rightarrow p$ is not a logical consequence of $p \rightarrow q$.

Exercise 28:
Prove that $\neg pimp \neg q$ is not a logical consequence of $p \rightarrow q$.

Exercise 29:
Prove that $\neg q \rightarrow \neg p$ is a logical consequence of $p \rightarrow q$.

Exercise 30:
Prove the following logical consequences

(1) $p \models p \vee q$
(2) $q \vee p \models p \vee q$
(3) $p \vee q, p \rightarrow r, q \rightarrow r \models r$
(4) $p \rightarrow q, p \models q$
(5) $p, \neg p \models q$

Solution

(1) Suppose that $\mathcal{I} \models p$, then by definition $\mathcal{I} \models p \vee q$.
(2) Suppose that $\mathcal{I} \models q \vee p$, then either $\mathcal{I} \models q$ or $\mathcal{I} \models p$. In both cases we have that $\mathcal{I} \models p \vee q$.
(3) Suppose that $\mathcal{I} \models p \vee q$ and $\mathcal{I} \models p \rightarrow r$ and $\mathcal{I} \models q \rightarrow r$. Then either $\mathcal{I} \models p$ or $\mathcal{I} \models q$. In the first case, since $\mathcal{I} \models p \rightarrow r$, then $\mathcal{I} \models r$, In the second case, since $\mathcal{I} \models q \rightarrow r$, then $\mathcal{I} \models r$.
(4) Suppose that $\mathcal{I} \models \neg p$, then not $\mathcal{I} \models p$, which implies that there is no $\mathcal{I}$ such that $\mathcal{I} \models p$ and $\mathcal{I} \models \neg p$. This implies that all the interpretations that satisfy p and $\neg p$ (actually none) satisfy also q.

□

9.5. Logical equivalence. Exercise 31:

Show that $\wedge$ is associative, i.e., $a \wedge (b \wedge c)$ is equivalent to $a \wedge (b \wedge c)$, and therefore one can write $a \wedge b \wedge c \wedge d \wedge \ldots$ without parentesis.

Solution Let us generate the truth table for both]

a	b	c	a	∧	(b	∧	c)	(a	∧	b)	∧	c
T	T	T	T	T	T	T	T	T	T	T	T	T
T	T	F	T	F	T	F	F	T	T	T	F	F
T	F	T	T	F	F	F	T	T	F	F	F	T
T	F	F	T	F	F	F	F	T	F	F	F	F
F	T	T	F	F	T	T	T	F	F	T	F	T
F	T	F	F	F	T	F	F	F	F	T	F	F
F	F	T	F	F	F	F	T	F	F	F	F	T
F	F	F	F	F	F	F	F	F	F	F	F	F

The truth values of the fomrulas are shown in the two columns blue and red. Notice that the two columns are identical. Therefore the two formulas are equivalent. ☐

Exercise 32:

Show that that $\to$ is a non associative operator, i.e., thjat $a \to (b \to c)$ is not equivalent to $(a \to b) \to c$.

Solution Let us build the truth table for the two formulas.

a	b	c	a	→	(b	→	c)	(a	→	b)	→	c
T	T	T	T	T	T	T	T	T	T	T	T	T
T	T	F	T	F	T	F	F	T	T	T	F	F
T	F	T	T	T	F	T	T	T	F	F	T	T
T	F	F	T	T	F	T	F	T	F	F	T	F
F	T	T	F	T	T	T	T	F	T	T	T	T
F	T	F	F	T	T	F	F	F	T	T	F	F
F	F	T	F	T	F	T	T	F	T	F	T	T
F	F	F	F	T	F	T	F	F	T	F	F	F

Notice that there are two assignments (highlighted in red), to a, b, c for which the truth value of the two formulas are not the same. The two formulas therefore are not equivalent. This implies that the expression $a \to b \to c$ is ambigous, and one should add the parenthesis in order to specify the correct parsing. In absence of parentesis the parsing $a \to (b \to c)$ is usually taken as correct. ☐

Exercise 33:

Repeat the previous exercise for the connective $\vee$.

Exercise 34:

Show that the $\leftrightarrow$ is commutative and associative.

Exercise 35:

Rewrite the following formulas by using only $\wedge$ and $\neg$.

(1) $p \vee q$
(2) $p \to q$
(3) $p \leftrightarrow q$
(4) $p \to (q \to r)$
(5) $p \leftrightarrow (q \vee r)$

Exercise 36:

Rewrite the following formulas by using only $\vee$ and $\neg$.

(1) $p \wedge q$
(2) $p \to q$
(3) $p \leftrightarrow q$
(4) $p \to (q \to r)$
(5) $p \leftrightarrow (q \wedge r)$

Bibliography

Brewka, Gerhard (1989). "Nonmonotonic Logics–A Brief Overview". In: *AI Communications* 2.2, pp. 88–97.
McCarthy, John (1959). "Programs with Common Sense". In: pp. 77–84.