

# Languages and Algorithms for Artificial Intelligence (Module 2)

Last update: 10 November 2023

Academic Year 2023 – 2024  
Alma Mater Studiorum · University of Bologna

# Contents

<b>1</b>	<b>Propositional logic</b>	<b>1</b>
1.1	Syntax . . . . .	1
1.2	Semantics . . . . .	1
1.2.1	Normal forms . . . . .	2
1.3	Reasoning . . . . .	3
1.3.1	Natural deduction . . . . .	4
<b>2</b>	<b>First order logic</b>	<b>5</b>
2.1	Syntax . . . . .	5

# 1 Propositional logic

## 1.1 Syntax

**Syntax** Rules and symbols to define well-formed sentences.

Syntax

The symbols of propositional logic are:

**Proposition symbols**  $p_0, p_1, \dots$

**Connectives**  $\wedge \vee \rightarrow \leftrightarrow \neg \perp ( )$

**Well-formed formula** The definition of a well-formed formula is recursive:

Well-formed formula

- An atomic proposition is a well-formed formula.
- If  $S$  is well-formed,  $\neg S$  is well-formed.
- If  $S_1$  and  $S_2$  are well-formed,  $S_1 \wedge S_2$  is well-formed.
- If  $S_1$  and  $S_2$  are well-formed,  $S_1 \vee S_2$  is well-formed.

Note that the implication  $S_1 \rightarrow S_2$  can be written as  $\neg S_1 \vee S_2$ .

The BNF definition of a formula is:

$$F := \text{atomic\_proposition} \mid F \wedge F \mid F \vee F \mid F \rightarrow F \mid F \leftrightarrow F \mid \neg F \mid (F)$$

## 1.2 Semantics

**Semantics** Rules to associate a meaning to well-formed sentences.

Semantics

**Model theory** What is true.

**Proof theory** What is provable.

**Interpretation** Given a propositional formula  $F$  of  $n$  atoms  $\{A_1, \dots, A_n\}$ , an interpretation  $\mathcal{I}$  of  $F$  is a pair  $(D, I)$  where:

Interpretation

- $D$  is the domain. Truth values in the case of propositional logic.
- $I$  is the interpretation mapping that assigns to the atoms  $\{A_1, \dots, A_n\}$  an element of  $D$ .

Note: given a formula  $F$  of  $n$  distinct atoms, there are  $2^n$  distinct interpretations.

**Model** If  $F$  is true under the interpretation  $\mathcal{I}$ , we say that  $\mathcal{I}$  is a model of  $F$  ( $\mathcal{I} \models F$ ).

Model

**Valid formula** A formula  $F$  is valid (tautology) iff it is true in all the possible interpretations. It is denoted as  $\models F$ .

Valid formula

**Invalid formula** A formula  $F$  is invalid iff it is not valid ( $\not\models$ ).

Invalid formula

In other words, there is at least an interpretation where  $F$  is false.

**Inconsistent formula** A formula  $F$  is inconsistent (unsatisfiable) iff it is false in all the possible interpretations. Inconsistent formula

**Consistent formula** A formula  $F$  is consistent (satisfiable) iff it is not inconsistent. Consistent formula  
In other words, there is at least an interpretation where  $F$  is true.

**Decidability** A logic is decidable if there is a terminating method to decide if a formula is valid. Decidability  
Propositional logic is decidable.

**Truth table** Useful to define the semantics of connectives. Truth table

- $\neg S$  is true iff  $S$  is false.
- $S_1 \wedge S_2$  is true iff  $S_1$  is true and  $S_2$  is true.
- $S_1 \vee S_2$  is true iff  $S_1$  is true or  $S_2$  is true.
- $S_1 \rightarrow S_2$  is true iff  $S_1$  is false or  $S_2$  is true.
- $S_1 \leftrightarrow S_2$  is true iff  $S_1 \rightarrow S_2$  is true and  $S_1 \leftarrow S_2$  is true.

**Evaluation** The connectives of a propositional formula are evaluated in the order: Evaluation order

$$\leftrightarrow, \rightarrow, \vee, \wedge, \neg$$

Formulas in parenthesis have higher priority.

**Logical consequence** Let  $\Gamma = \{F_1, \dots, F_n\}$  be a set of formulas (premises) and  $G$  a formula (conclusion).  $G$  is a logical consequence of  $\Gamma$  ( $\Gamma \models G$ ) if in all the possible interpretations  $\mathcal{I}$ , if  $F_1 \wedge \dots \wedge F_n$  is true,  $G$  is true. Logical consequence

**Logical equivalence** Two formulas  $F$  and  $G$  are logically equivalent ( $F \equiv G$ ) iff the truth values of  $F$  and  $G$  are the same under the same interpretation. In other words,  $F \equiv G \iff F \models G \wedge G \models F$ . Logical equivalence

Common equivalences are:

**Commutativity** :  $(P \wedge Q) \equiv (Q \wedge P)$  and  $(P \vee Q) \equiv (Q \vee P)$

**Associativity** :  $((P \wedge Q) \wedge R) \equiv (P \wedge (Q \wedge R))$  and  $((P \vee Q) \vee R) \equiv (P \vee (Q \vee R))$

**Double negation elimination** :  $\neg(\neg P) \equiv P$

**Contraposition** :  $(P \rightarrow Q) \equiv (\neg Q \rightarrow \neg P)$

**Implication elimination** :  $(P \rightarrow Q) \equiv (\neg P \vee Q)$

**Biconditional elimination** :  $(P \leftrightarrow Q) \equiv ((P \rightarrow Q) \wedge (Q \rightarrow P))$

**De Morgan** :  $\neg(P \wedge Q) \equiv (\neg P \vee \neg Q)$  and  $\neg(P \vee Q) \equiv (\neg P \wedge \neg Q)$

**Distributivity of  $\wedge$  over  $\vee$**  :  $(P \wedge (Q \vee R)) \equiv ((P \wedge Q) \vee (P \wedge R))$

**Distributivity of  $\vee$  over  $\wedge$**  :  $(P \vee (Q \wedge R)) \equiv ((P \vee Q) \wedge (P \vee R))$

### 1.2.1 Normal forms

**Negation normal form (NNF)** A formula is in negation normal form iff negations appear only in front of atoms (i.e. not parenthesis). Negation normal form

**Conjunctive normal form (CNF)** A formula  $F$  is in conjunctive normal form iff: Conjunctive normal form

- it is in negation normal form;

- it has the form  $F := F_1 \wedge F_2 \cdots \wedge F_n$ , where each  $F_i$  (clause) is a disjunction of literals.

**Example.**

$(\neg P \vee Q) \wedge (\neg P \vee R)$  is in CNF.

$\neg(P \vee Q) \wedge (\neg P \vee R)$  is not in CNF (not in NNF).

**Disjunctive normal form (DNF)** A formula  $F$  is in disjunctive normal form iff:

Disjunctive normal form

- it is in negation normal form;
- it has the form  $F := F_1 \vee F_2 \cdots \vee F_n$ , where each  $F_i$  is a conjunction of literals.

## 1.3 Reasoning

**Reasoning method** Systems to work with symbols.

Reasoning method

Given a set of formulas  $\Gamma$ , a formula  $F$  and a reasoning method  $E$ , we denote with  $\Gamma \vdash^E F$  the fact that  $F$  can be deduced from  $\Gamma$  using the reasoning method  $E$ .

**Sound** A reasoning method  $E$  is sound iff:

Soundness

$$(\Gamma \vdash^E F) \rightarrow (\Gamma \models F)$$

**Complete** A reasoning method  $E$  is complete iff:

Completeness

$$(\Gamma \models F) \rightarrow (\Gamma \vdash^E F)$$

**Deduction theorem** Given a set of formulas  $\{F_1, \dots, F_n\}$  and a formula  $G$ :

Deduction theorem

$$(F_1 \wedge \cdots \wedge F_n) \models G \iff \models (F_1 \wedge \cdots \wedge F_n) \rightarrow G$$

*Proof.*

$\rightarrow$  ) By hypothesis  $(F_1 \wedge \cdots \wedge F_n) \models G$ .

So, for each interpretation  $\mathcal{I}$  in which  $(F_1 \wedge \cdots \wedge F_n)$  is true,  $G$  is also true. Therefore,  $\mathcal{I} \models (F_1 \wedge \cdots \wedge F_n) \rightarrow G$ .

Moreover, for each interpretation  $\mathcal{I}'$  in which  $(F_1 \wedge \cdots \wedge F_n)$  is false,  $(F_1 \wedge \cdots \wedge F_n) \rightarrow G$  is true. Therefore,  $\mathcal{I}' \models (F_1 \wedge \cdots \wedge F_n) \rightarrow G$ .

In conclusion,  $\models (F_1 \wedge \cdots \wedge F_n) \rightarrow G$ .

$\leftarrow$  ) By hypothesis  $\models (F_1 \wedge \cdots \wedge F_n) \rightarrow G$ . Therefore, for each interpretation where  $(F_1 \wedge \cdots \wedge F_n)$  is true,  $G$  is also true.

In conclusion,  $(F_1 \wedge \cdots \wedge F_n) \models G$ .

□

**Refutation theorem** Given a set of formulas  $\{F_1, \dots, F_n\}$  and a formula  $G$ :

Refutation theorem

$$(F_1 \wedge \cdots \wedge F_n) \models G \iff F_1 \wedge \cdots \wedge F_n \wedge \neg G \text{ is inconsistent}$$

Note: this theorem is not accepted in intuitionistic logic.

*Proof.* By definition,  $(F_1 \wedge \cdots \wedge F_n) \models G$  iff for every interpretation where  $(F_1 \wedge \cdots \wedge F_n)$  is true,  $G$  is also true. This requires that there are no interpretations where  $(F_1 \wedge \cdots \wedge F_n)$  is true and  $G$  false. In other words, it requires that  $(F_1 \wedge \cdots \wedge F_n \wedge \neg G)$  is inconsistent. □

### 1.3.1 Natural deduction

**Proof theory** Set of rules that allows to derive conclusions from premises by exploiting syntactic manipulations. Proof theory

**Natural deduction** Set of rules to introduce or eliminate connectives. We consider a subset  $\{\wedge, \rightarrow, \perp\}$  of functionally complete connectives. Natural deduction for propositional logic

Natural deduction can be represented using a tree like structure:

$$\begin{array}{c} [\text{hypothesis}] \\ \vdots \\ \frac{\text{premise}}{\text{conclusion}} \text{ rule name} \end{array}$$

The conclusion is true when the hypothesis are able to prove the premise. Another tree can be built on top of premises to prove them.

**Introduction** Usually used to prove the conclusion by splitting it. Introduction rules

$$\begin{array}{c} [\varphi] \\ \vdots \\ \frac{\psi \quad \varphi}{\varphi \wedge \psi} \wedge\text{I} \end{array} \qquad \begin{array}{c} [\varphi] \\ \vdots \\ \frac{\psi}{\varphi \rightarrow \psi} \rightarrow\text{I} \end{array}$$

**Elimination** Usually used to exploit hypothesis and derive a conclusion. Elimination rules

$$\frac{\varphi \wedge \psi}{\varphi} \wedge\text{E} \qquad \frac{\varphi \wedge \psi}{\psi} \wedge\text{E} \qquad \frac{\varphi \quad \varphi \rightarrow \psi}{\psi} \rightarrow\text{E}$$

**Ex falso sequitur quodlibet** From contradiction, anything follows. This can be used when we have two contradicting hypothesis. Ex falso sequitur quodlibet

$$\frac{\perp}{\varphi} \perp$$

**Reductio ad absurdum** Assume the opposite and prove a contradiction (not accepted in intuitionistic logic). Reductio ad absurdum

$$\begin{array}{c} [\neg\varphi] \\ \vdots \\ \frac{\perp}{\varphi} \text{ RAA} \end{array}$$

## 2 First order logic

### 2.1 Syntax

The symbols of propositional logic are:

Syntax

**Constants** Known elements of the domain. Do not represent truth values.

**Variables** Unknown elements of the domain. Do not represent truth values.

**Function symbols** Function  $f^{(n)}$  applied on  $n$  constants to obtain another constant.

**Predicate symbols** Function  $P^{(n)}$  applied on  $n$  constants to obtain a truth value.

**Connectives**  $\forall \exists \wedge \vee \rightarrow \neg \leftrightarrow \perp ( )$

Using the basic syntax, the following constructs can be defined:

**Term** Denotes elements of the domain.

$$t := \text{constant} \mid \text{variable} \mid f^{(n)}(t_1, \dots, t_n)$$

**Proposition** Denotes truth values.

$$P := \perp \mid P \wedge P \mid P \vee P \mid P \rightarrow P \mid P \leftrightarrow P \mid \neg P \mid \forall x.P \mid \exists x.P \mid (P) \mid P^{(n)}(t_1, \dots, t_n)$$

**Well-formed formula** The definition of well-formed formula in first order logic extends the one of propositional logic by adding the following conditions:

Well-formed formula

- If  $S$  is well-formed,  $\exists X.S$  is well-formed. Where  $X$  is a variable.
- If  $S$  is well-formed,  $\forall X.S$  is well-formed. Where  $X$  is a variable.