# LOGIC

Basic concept: atomic **proposition** (or **statement**): either **true** or **false** independent of context. **true** and **false** are called **truth values**. Compound propositions consist of atomic propositions and linguistic connectors corresponding to Boolean operators.

We shall study two models.

**I. Propositional logic (other names: propositional calculus, zeroth-order logic)**

The formulas of the logic are built from atomic propositions, which are statements that have *no internal structure*. Formulas can be combined using Boolean operators (and, or, etc.). Atomic propositions can be either true or false.

*Advantage:* simple, close to natural languages

*Disadvantage:* not sufficiently expressive for formalizing mathematical theories

*An application:* digital circuits

**II. First-order logic (other names: predicate logic, predicate calculus)**

Statements have inner structure. Allows the use of sentences that contain variables and symbols that can be interpreted as functions and relations. Variables range over a domain. Formulas can be combined using Boolean operators (and, or, etc.) and quantifiers.

*Advantage:* more expressive power can describe the world more accurately

*Disadvantage:* more difficult

*An application:* automated theorem proving (Prolog language)

## Propositional logic

### I. Syntax

Let $\mathcal{P} = \{p, q, r, \ldots\}$ be a (countably) infinite set, its elements are called **atoms** (also called: atomic propositions, variables, atomic variables).
We define an **alphabet** (the set of terminals) as follows:

- elements of $\mathcal{P}$,
- Boolean operators:

| | | | |
|---|---|---|---|
| negation | $\neg$ | conjunction | $\wedge$ |
| disjunction | $\vee$ | implication | $\rightarrow$ |

- ( , ) (in the string representation only).

Note: sometimes further Boolean operators are used as well, such as equivalence ($\leftrightarrow$), exclusive or ($\oplus$), nor ($\downarrow$), nand ($\uparrow$).

**Formula:** A formula is one of the following rooted node-labelled binary trees

- a single node (root) labelled by an atom
- a node (root) labelled by $\neg$ with a single child that is a formula
- a node (root) labelled by one of the binary operations with two children both of which are formulas

**Subformula** A (proper) subformula is a (proper) subtree.

**Scope of an operator** The subformula having the given operator in its root.

**Principal operator of a formula** The operator at the root of the tree. (I.e., its scope is the whole formula.)

(Formulas, that are atoms have no principal operator).

**Algorithm (for producing the string representation)**
Input: a formula $F$ given by its tree representation
Output: string representation of $F$

**Algorithm 1** INORDER($F$)

```
if F is a leaf then
    write its label
    return
let F_1 and F_2 be the left and right subtrees of F
write a left parenthesis '('
INORDER(F_1)
write the label of the root of F
INORDER(F_2)
write a right parenthesis ')'
```

If the root of $F$ is labelled by $\neg$, the left subtree is considered to be empty and the step INORDER($F_1$) is skipped.

**Leaving parenthesis**

Analogy: arithmetic expression, e.g., $3 + 8 \cdot 7$.

**Order of precedence** from high to low: $\neg$, $\wedge$, $\vee$, $\rightarrow$.

Operators are assumed to associate to the right, that is, $A \vee B \vee C$ means $(A \vee (B \vee C))$.

Parentheses are used only if they are needed to indicate an order different from that imposed by the rules for precedence and associativity.

## II. Semantics

### Interpretation

Analogy: arithmetic expressions. E.g., $y = a + 2 \cdot b$. Assign values to $a$ and $b$ then evaluate $y$.

Let $A$ be a formula and let $P_A$ be the set of atoms appearing in $A$. An **interpretation** for $A$ is a total function $I : P_A \to \{T, F\}$ that assigns one of the truth values $T$ (true) or $F$ (false) to every atom in $P_A$.

### Truth value of a formula

Let $I$ be an interpretation for a formula $A$. $v_I(A)$, the **truth value of $A$ under $I$** is defined recursively as follows:

| | |
|---|---|
| $v_I(A) = I(A)$ | if $A$ is an atom |
| $v_I(\neg A) = T$ | if $v_I(A) = F$ |
| $v_I(\neg A) = F$ | if $v_I(A) = T$ |
| $v_I(A_1 \wedge A_2) = T$ | if $v_I(A_1) = T$ and $v_I(A_2) = T$ |
| $v_I(A_1 \wedge A_2) = F$ | in the other three cases |
| $v_I(A_1 \vee A_2) = F$ | if $v_I(A_1) = F$ and $v_I(A_2) = F$ |
| $v_I(A_1 \vee A_2) = T$ | in the other three cases |
| $v_I(A_1 \to A_2) = F$ | if $v_I(A_1) = T$ and $v_I(A_2) = F$ |
| $v_I(A_1 \to A_2) = T$ | in the other three cases |

A **truth table** for a formula $A$ is a table with $n + 1$ columns and $2^n$ rows, where $n = |P_A|$. There is a column for each atom in $P_A$ , plus a column for the formula $A$. The first $n$ columns specify the interpretation $I$ that maps atoms in $P_A$ to $\{T, F\}$. The last column shows $v_I(A)$, the truth value of $A$ for the interpretation $I$.

### Semantic properties

- $A$ is **satisfiable** iff $v_I(A) = T$ for some interpretation $I$ .
- A satisfying interpretation $I$ is a **model** for $A$, denoted $I \models A$.
- $A$ is **valid**, denoted $\models A$, iff $v_I(A) = T$ for all interpretations $I$ . A valid propositional formula is also called a **tautology**.
- $A$ is **unsatisfiable** iff it is not satisfiable, that is, if $v_I(A) = F$ for all interpretations $I$.
- $A$ is **falsifiable**, denoted $\not\models A$, iff it is not valid, that is, if $v_I(A) = F$ for some interpretation $I$.
- Let $A_1, A_2$ be formulas. If $v_I(A_1) = v_I(A_2)$ for all interpretations $I$ , then $A_1$ is **logically equivalent** to $A_2$ , denoted $A_1 \equiv A_2$.

## Some laws of propositional logic

$\top$: constant valid  $\perp$: constant unsatisfiable

- $A \vee \top \equiv \top$ and $A \wedge \perp \equiv \perp$,
- $A \vee \perp \equiv A$ and $A \wedge \top \equiv A$.
- $A \vee \neg A \equiv \top$ and $A \wedge \neg A \equiv \perp$,
- $\neg\neg A \equiv A$,
- $A \vee A \equiv A$ and $A \wedge A \equiv A$ (idempotence),
- $A \to B \equiv \neg A \vee B$,
- $A \to B \equiv \neg B \to \neg A$ (rhs: contrapositive of the lhs).

- $A \vee B \equiv B \vee A$ and $A \wedge B \equiv B \wedge A$ (commutative laws),
- $(A \vee B) \vee C \equiv A \vee (B \vee C)$ and $(A \wedge B) \wedge C \equiv A \wedge (B \wedge C)$ (associative laws),
- $(A \vee B) \wedge C \equiv (A \wedge C) \vee (B \wedge C)$ and $(A \wedge B) \vee C \equiv (A \vee C) \wedge (B \vee C)$ (distributive laws),
- $\neg(A \wedge B) \equiv \neg A \vee \neg B$ and $\neg(A \vee B) \equiv \neg A \wedge \neg B$ (De Morgan laws),
- $(A \vee B) \wedge B \equiv B$ and $(A \wedge B) \vee B \equiv B$.

- $A \leftrightarrow B \equiv (A \to B) \wedge (B \to A)$,
- $A \oplus B \equiv (A \vee B) \wedge \neg(A \wedge B)$,
- $A \uparrow B \equiv \neg(A \wedge B)$,
- $A \downarrow B \equiv \neg(A \vee B)$.

**Definition** A **literal** is either an atom or a negation of an atom.

**Definition** A formula is in **conjunctive normal form (CNF)** iff it is a conjunction of disjunctions of literals.

**Example:**

| | |
|---|---|
| $(\neg p \vee q \vee r) \wedge (\neg q \vee r) \wedge (\neg r)$ | CNF |
| $(\neg p \vee q \vee r) \wedge ((p \wedge \neg q) \vee r) \wedge (\neg r)$ | not in CNF |
| $(\neg p \vee q \vee r) \wedge \neg(\neg q \vee r) \wedge (\neg r)$ | not in CNF |
| $\neg p \vee q \vee r$ | CNF |

**Theorem** For every formula $A$ in propositional logic there is a logically equivalent formula in CNF.

**Notation** Another representation of a CNF is **clausal form**:

- A *clause* is a set of literals. Example: $\{\neg q, \neg p, q\}$.
- A clause is considered to be an implicit disjunction of its literals. Example: $\{\neg q, \neg p, q\}$ is $\neg q \vee \neg p \vee q$.
- A *unit clause* is a clause consisting of exactly one literal. Example: $\{\neg q\}$.
- The empty set of literals is the *empty clause*, denoted by $\square$.
- A formula in *clausal form* is a set of clauses. Example: $\{\{p, r\}, \{\neg q, \neg p, q\}\}$.
- A formula is considered to be an implicit conjunction of its clauses. Example: $(p \vee r) \wedge (\neg q \vee \neg p \vee q)$ for the previous one.
- The formula that is the empty set of clauses is denoted by $\emptyset$.

**Proposition:** Every formula in propositional logic can be transformed into an logically equivalent formula in clausal form.

multiple occurrences of literals and clauses $\Rightarrow$ single occurance
equivalent because of idempotence ($A \vee A \equiv A,\ A \wedge A \equiv A$)

Example:

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

Clausal form: $\{\{p, r\}, \{\neg q, \neg p, q\}, \{p, \neg p, q\}\}$

A clause if *trivial* if it contains a pair of clashing literals.

**Proposition:** Let $S$ be a set of clauses and let $C \in S$ be a trivial clause. Then $S - \{C\}$ is logically equivalent to $S$. (True, because of $A \vee \top \equiv \top$ and $A \wedge \top \equiv A$. So we can delete trivial clauses.)

**Proposition** $\square$ (empty clause) is unsatisfiable. $\emptyset$ (the empty set of clauses) is valid.

**Notation** The set delimiters { and } are removed from each clause and a negated literal is denoted by a bar over the atomic proposition.

**Notation**: Let CNF($A$) and cf($A$) denote a CNF and a clausal form for a formula $A$, respectively.

**Example:**

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

$\text{CNF}(A) = (p \vee r) \wedge (\neg q \vee \neg p \vee q) \wedge (p \vee \neg p \vee q \vee p \vee \neg p) \wedge (r \vee p)$

$\text{cf}(A) = \{\{p, r\}, \{\neg q, \neg p, q\}, \{p, \neg p, q\}\}$ which becomes

$\text{cf}(A) = \{pr, \bar{q}\bar{p}q, p\bar{p}q\}$ using this shorter notation.

**Theorem** Let $U = \{A_1, \ldots A_n\}$ be a finite set of formulas and let $B$ be a formula. Then the following statements are equivalent.

- $\{A_1, \ldots A_n\} \models B$
- $\{A_1 \wedge \cdots \wedge A_n\} \models B$
- $A_1 \wedge \cdots \wedge A_n \wedge \neg B$ unsatisfiable
- $\{A_1, \ldots, A_n, \neg B\}$ unsatisfiable
- $\{\text{CNF}(A_1), \ldots, \text{CNF}(A_n), \text{CNF}(\neg B)\}$ unsatisfiable
- $\text{cf}(A_1) \cup \cdots \cup \text{cf}(A_n) \cup \text{cf}(\neg B)$ unsatisfiable

If $\ell$ is a literal, let $\ell^c$ denote its complementary pair.
**Definition** Let $C_1, C_2$ be clauses such that $\ell \in C_1, \ell^c \in C_2$. The clauses $C_1, C_2$ are said to be **clashing clauses** and to **clash on the complementary pair of literals** $\ell, \ell^c$ . $C$, the **resolvent** of $C_1$ and $C_2$ , is the clause:

$$\text{Res}(C_1, C_2) = (C_1 - \{\ell\}) \cup (C_2 - \{\ell^c\}).$$

$C_1$ and $C_2$ are the parent clauses of $C$.

**Example:** $C_1 = ab\bar{c}$ and $C_2 = bc\bar{e}$ They clash on the pair of complementary literals $c, \bar{c}$.

$\text{Res}(C_1, C_2) = (ab\bar{c} - \{\bar{c}\}) \cup (bc\bar{e} - \{c\}) = ab \cup b\bar{e} = ab\bar{e}$.
Let $\binom{S}{2}$ denote the set of 2-element subsets of $S$.

**Algorithm 2** RESOLUTION PROCEDURE($S$)

**while** there is an unmarked pair of $\binom{S}{2}$ **do**
    choose an unmarked pair $\{C_1, C_2\}$ of $\binom{S}{2}$ and mark it
    **if** $\{C_1, C_2\}$ is a clashing pair of clauses **then**
        $C \leftarrow \text{Res}(C_1, C_2)$
        **if** $C = \square$ **then**
            **return** '$S$ is unsatisfiable'
        **else**
            **if** $C$ is not the trivial clause **then**
                $S \leftarrow S \cup \{C\}$
**return** '$S$ is satisfiable'

**Theorem** Resolution procedure is sound and complete.

**III. Exercises**

Exercise: Is it a propsition or not.

1. 2 times 2 equals 5. (In the ring of natural numbers.)
2. I will write my homework tomorrow.
3. The monarch of USA visited London last week.
4. This statement is false.
5. The teacher of our school is 50 years old.
6. What is the sense of this exercise?

Exercise: Formalize the sentences.

1. Alice is not dancing with Bob unless Bob invites her for a Coke.
2. I go to the cinema with you but i have to finish my cooking first.
3. Prospering business needs good planing.
4. A practice mark is needed to register for the exam.
5. Bob invites Alice for a Coke only if Alice smiles at him first.

Exercise: Leave as many paranthesis as you can.

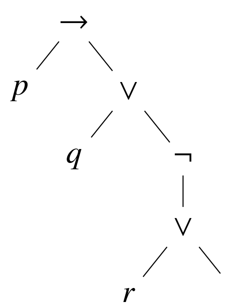
1. $(p \to (q \vee (\neg(r \vee p))))$
2. $(((p \to q) \vee (\neg r)) \vee p)$

Solution:

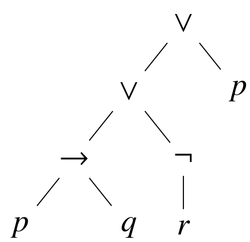
1. $p \to q \vee \neg(r \vee p)$
2. $((p \to q) \vee \neg r) \vee p$

Exercise: Show an example for the need of paranthesis.

Solution:



String representation:
$(p \to (q \lor (\neg(r \lor p))))$



String representation:
$(((p \to q) \lor (\neg r)) \lor p)$

Exercise: Make the truth table for $A = p \to q \lor \neg(r \lor p)$.

Solution:

| $p$ | $q$ | $r$ | $r \lor p$ | $\neg(r \lor p)$ | $q \lor \neg(r \lor p)$ | $p \to q \lor \neg(r \lor p)$ |
|---|---|---|---|---|---|---|
| $T$ | $T$ | $T$ | $T$ | $F$ | $T$ | $T$ |
| $T$ | $T$ | $F$ | $T$ | $F$ | $T$ | $T$ |
| $T$ | $F$ | $T$ | $T$ | $F$ | $F$ | $F$ |
| $T$ | $F$ | $F$ | $T$ | $F$ | $F$ | $F$ |
| $F$ | $T$ | $T$ | $T$ | $F$ | $T$ | $T$ |
| $F$ | $T$ | $F$ | $F$ | $T$ | $T$ | $T$ |
| $F$ | $F$ | $T$ | $T$ | $F$ | $F$ | $T$ |
| $F$ | $F$ | $F$ | $F$ | $T$ | $T$ | $T$ |

Exercise: Give examples for the semantic properties.

Solution:

- Let $A = p \to q \lor \neg(r \lor p)$.

  Then $A$ is satisfiable, falsifiable, interpretation $TTT$ is a model for $A$. (See its truth table.)

  On the other hand $A$ is not valid and not unsatisfiable.

- formula $p \lor \neg p$ is valid, on the other hand $p \land \neg p$ is unsatisfiable.

- $p \lor q$ and $q \lor p$ are logically equivalent formulas.

**Exercise:** Prove!

1. $(x \to y) \lor (y \to z) \equiv x \to (y \lor z)$
2. $\neg(x \lor (y \land (z \to x))) \equiv \neg x \land (y \to z)$
3. $\models (x \to y \to z) \to (x \to y) \to x \to z$,
4. $\models x \to y \to x \land y$.

**Exercise:** Let $A = (p \to q) \to r$. Give a formula $B$ in CNF satisfying $A \equiv B$.
**Solution:** The truth table for $A$ is the following

| $p$ | $q$ | $r$ | $A$ |
|---|---|---|---|
| $T$ | $T$ | $T$ | $T$ |
| $T$ | $T$ | $F$ | $F$ |
| $T$ | $F$ | $T$ | $T$ |
| $T$ | $F$ | $F$ | $T$ |
| $F$ | $T$ | $T$ | $T$ |
| $F$ | $T$ | $F$ | $F$ |
| $F$ | $F$ | $T$ | $T$ |
| $F$ | $F$ | $F$ | $F$ |

$(\neg p \vee \neg q \vee r) \wedge (p \vee \neg q \vee r) \wedge (p \vee q \vee r)$ is a CNF equivalent with $A$.

**Exercise** Make an equivalent CNF

1. $(x \to y) \to \neg(x \wedge \neg y)$
2. $\neg(x \to z) \vee \neg(\neg x \vee z \to y)$
3. $\neg(\neg x \vee z \to y \vee x)$

**Exercise:** Prove that $\{q \vee r,\ \neg p \vee s \vee \neg r,\ \neg q,\ q \vee \neg r \vee \neg s,\ p \vee q\}$ is unsatisfiable.

**Solution:**
$\mathcal{S} = \{q \vee r,\ \neg p \vee s \vee \neg r,\ \neg q,\ q \vee \neg r \vee \neg s,\ p \vee q\}$

1. $\neg q$ $\quad (\in \mathcal{S})$
2. $q \vee r$ $\quad (\in \mathcal{S})$
3. $r$ $\quad (= \mathrm{res}(1,2))$
4. $\neg p \vee s \vee \neg r$ $\quad (\in \mathcal{S})$
5. $q \vee \neg r \vee \neg s$ $\quad (\in \mathcal{S})$
6. $\neg p \vee q \vee \neg r$ $\quad (= \mathrm{res}(4,5))$
7. $\neg p \vee q$ $\quad (= \mathrm{res}(3,6))$
8. $p \vee q$ $\quad (\in \mathcal{S})$
9. $q$ $\quad (= \mathrm{res}(7,8))$
10. $\Box$ $\quad (= \mathrm{res}(1,9))$

**Exercise:**

(1) A1 If we go to Madrid we visit Barcelona and Sevilla as well.

A2 If we do not visit Sevilla, then we visit Barcelona.

A3 If we visit Sevilla, then we visit Madrid as well.

B So we visit Barcelona.

(2) A1 If the line has no common point with the plain, then they are parallel.

A2 If the line has more than one common point with the plain, then the line is incident to the plain.

A3 The line is neither parallel with nor incident to the plain.

B So, the line has exactly one common point with the plain.

Prove that $\{A1, A2, A3\} \models B$ (formalize the sentences, make a set of clauses $S$ such that unsatisfiability of $S$ is equivalent with the consequence. Prove that $S$ is unsatisfiable).