

A **first-order language** L consists of the following data:

- (1) **logical symbols** consisting of
 - (i) **variables** x_0, x_1, x_2, \dots ,
 - (ii) **connectives** \neg (negation) and \vee (disjunction),
 - (iii) **quantifier** \exists (existential quantifier),
 - (iv) **equality symbol** $=$;
- (2) **nonlogical symbols**
 - (i) **constant symbols** $\{c_i : i \in I\}$ for any indexing set I ,
 - (ii) for all $n \in \mathbb{Z}_{>0}$, **n -ary function symbols** $\{f_j : j \in J_n\}$ for any indexing set J_n (constants may be considered as 0-ary functions),
 - (iii) for all $n \in \mathbb{Z}_{>0}$, **n -ary relation symbols** $\{R_k : k \in K_n\}$ for an indexing set K_n .

Let L be a first-order language. A **term** of L is defined inductively as follows:

- (i) any variable or constant is a term (of **rank** 0),
- (ii) if t_1, \dots, t_n are terms (of ranks k_1, \dots, k_n), respectively, and f_j is an n -ary function symbol, then $f_j t_1 \cdots t_n$ (or $f(t_1, \dots, t_n)$) is a term (of rank $\max\{k_1, \dots, k_n\} + 1$).

Let L be a first-order language. An **atomic formula** of L is one of the following:

- (i) $t = s$ whenever t and s are terms.
- (ii) $pt_1 \cdots t_n$ whenever p is an n -ary relation symbol and t_1, \dots, t_n are terms.

Let L be a first-order language. A **formula** of L is defined inductively as follows:

- (i) any atomic formula is a formula (of **rank** 0),
- (ii) if A is a formula, then $\neg A$ is a formula,
- (iii) if A and B are formulas, then $A \vee B$ is a formula,
- (iv) if A is a formula and x is a variable, then $\exists x A$ is a formula.

The set of all formulas of L is the smallest set of expressions/strings which contains the atomic formulas and is closed under negation, disjunction, and existential quantification.

For quality of life, we introduce a number of commonly used symbols, which in this context will function as abbreviations for formulas. Note that our definition of a first-order language could include these as "built-in" symbols, but we choose to implement them as syntax sugar for ease of use.

- $A \wedge B := \neg(\neg A \vee \neg B)$ (conjunction),
- $A \rightarrow B := \neg A \vee B$ (implication),
- $A \leftrightarrow B := (A \rightarrow B) \wedge (B \rightarrow A)$ (biconditional),
- $\forall x A := \neg \exists x \neg A$ (universal quantification).

A formula of the form $\exists x A$ is called an **instantiation** of A ,

A formula of the form $\forall x A$ is called a **generalization** of A .

A formula is called **elementary** if it is an atomic formula or an instantiation of a formula.

Let A be a formula. A **subformula** of A is defined inductively as follows:

1. A is a subformula of A ,
2. B is a subformula of A whenever $\neg B$ or $\exists x B$ is a subformula of A ,
3. B and C are subformulas of A whenever $B \vee C$ is a subformula of A .

An occurrence of a variable v in a formula A is **bound** if it occurs in a subformula of the form $\exists v B$ (specifically, if v occurs in B); otherwise, the occurrence is **free**.

(write $\varphi[v_1, \dots, v_n]$ if φ is a formula in which the free variables are among v_1, \dots, v_n)

A formula with no free variables is called a **sentence** or said to be **closed**.

A formula which contains no quantifiers (i.e., all variables occur free) is said to be **open**.

Let t be a term, v a variable, and A a formula of a language L . We say that t is **substitutable** for v in A if for each variable w occurring t there is no subformula of A of the form $\exists w B$ where v is free in B .

In which case, we write $A[t/v]$ (or $A_v[t]$) for the formula obtained by replacing each free occurrence of v in A by t .

A first-order theory T consists of the following data:

- (i) a first-order language L (or $L(T)$),
- (ii) a set of formulas of L called the **nonlogical axioms** of T .

Example. We define the theory N . It's nonlogical symbols are as follows:

- a constant symbol 0 ,
- a unary function symbol S (successor),
- binary function symbols $+$ (addition) and \cdot (multiplication),
- a binary relation symbol $<$ ("less than").

It's nonlogical axioms are as follows:

- (1) $\forall x(\neg(Sx = 0))$,
- (2) $\forall x\forall y(Sx = Sy \rightarrow x = y)$,
- (3) $\forall x(x + 0 = x)$,
- (4) $\forall x\forall y(x + Sy = S(x + y))$,
- (5) $\forall x(x \cdot 0 = 0)$,
- (6) $\forall x\forall y(x \cdot Sy = (x \cdot y) + x)$,
- (7) $\forall x(\neg(x < 0))$,
- (8) $\forall x\forall y(x < Sy \rightarrow (x < y \vee x = y))$,
- (9) $\forall x\forall y(x < y \vee x = y \vee y < x)$.

For any nonnegative integer n , we define the term $k_n := S \cdots S0$ with n occurrences of S . Such terms are called **numerals**. Note that $k_0 = 0$ is a constant.

Example. Peano Arithmetic (PA) is the theory obtained from N by removing axiom (9) and adding the **induction axiom schema**: for each formula $A[v]$ the formula

$$A[0/v] \rightarrow \forall v(A \rightarrow A[Sw]) \rightarrow A.$$

(why not $\cdots \rightarrow \forall v A[v]$? why is the conclusion a formula with free variables?)

We define **Zermelo-Fraenkel set theory (ZF)**. There are multiple equivalent formulations with slightly different sets of axioms, we provide one of these. It's nonlogical symbols are as follows:

- a binary relation symbol \in ("in" or "is an element of").

It's nonlogical axioms are as follows:

- (1) **Existence.** There exists a set. The formula

$$\exists x(x = x)$$

is an axiom. (This axiom is not necessary, as it can generally be deduced from the other axioms.)

- (2) **Extensionality.** Two sets are equal if they have the same elements. The formula

$$\forall x \forall y (\forall z (z \in x \leftrightarrow z \in y) \rightarrow x = y)$$

is an axiom.

- (3) **Comprehension/Specification Schema.** For each formula $\varphi[x, w_1, \dots, w_n]$ (in which y is not free) the formula

$$\forall z \forall w_1 \dots \forall w_n (\exists y \forall x (x \in y \leftrightarrow (x \in z \wedge \varphi[x, w_1, \dots, w_n])))$$

is an axiom. In other words, given a set z and a predicate of sets φ , there is a subset y of z which contains exactly the elements of z satisfying φ . Sometimes denote

$$y = \{x \in z : \varphi[x, w_1, \dots, w_n]\}.$$

- (4) **Replacement Schema.** For each formula $\varphi[x, y, z, u_1, \dots, u_n]$ the formula

$$\forall z \forall u_1 \dots \forall u_n (\forall (x \in z) \exists! y \varphi \rightarrow \exists v \forall x (x \in z \rightarrow \exists y (y \in v \wedge \varphi)))$$

is an axiom. Here, we use $\exists!$ to denote unique existence, i.e.,

$$\exists! y \varphi := \exists y (\varphi \wedge \forall u (\varphi[u/y] \rightarrow u = y)).$$

Also, $\forall (x \in z) A := \forall x (x \in z \rightarrow A)$. With comprehension, this says that the range of a function is a set.

- (5) **Pairing.** Given sets x and y there is a set containing both x and y . The formula

$$\forall x \forall y \exists z (x \in z \wedge y \in z)$$

is an axiom.

- (6) **Union.** Given a set x , there is a set containing all the elements of the elements of x . The formula

$$\forall x \exists y \forall z \forall u ((z \in x \wedge u \in z) \rightarrow u \in y)$$

is an axiom. Together with comprehension, this implies that the union of a set of sets is a set.

- (7) **Power Set.** Given a set x , there is a set containing all the subsets of x . The formula

$$\forall x \exists y \forall z (\forall u (u \in z \rightarrow u \in x) \rightarrow z \in y)$$

is an axiom.

(8) **Infinity.** There exists an infinite set. The formula

$$\exists x(\emptyset \in x \wedge \forall y(y \in x \rightarrow (y \cup \{y\}) \in x))$$

It can be shown that the **empty set** \emptyset exists.

(9) **Foundation/Regularity.** Every nonempty set has an \in -minimal element. The formula

$$\forall x(\exists y(y \in x) \rightarrow \exists y(y \in x \wedge \neg \exists z(z \in x \wedge z \in y)))$$

is an axiom.

A **structure/interpretation** of a first-order language L consists of the following data:

- (i) a nonempty set M called the **domain/universe** of the structure,
- (ii) for each constant symbol c of L a fixed element $c_M \in M$,
- (iii) for each n -ary function symbol f of L an n -ary map $f_M : M^n \rightarrow M$,
- (iv) for each n -ary relation symbol p of L an n -ary relation $p_M \subseteq M^n$ on M .

The interpretation of “=” is always taken to be the equality relation on M .

Example. Let \mathbb{N} be the set of natural numbers, where $0, 1, +, \cdot, <$ have their usual meanings. Define the successor function as $S(n) = n + 1$ for all $n \in \mathbb{N}$. Then \mathbb{N} is a structure of the language $L(\mathbb{N})$, which we take to be the standard structure of N .

Let M be a structure of L . For each variable-free term t of L , we define the **interpretation/meaning** t_M of t in M inductively on the rank of t :

- c_M is automatic for constants c ,
- $(ft_1 \cdots t_n)_M := f_M((t_1)_M, \dots, (t_n)_M)$ for n -ary function symbols f and variable-free terms t_1, \dots, t_n .

Let L be a first-order language with universe M .

Define L_M to be the first-order language obtained from L by adding a constant symbol i_a for each $a \in M$. The symbol i_a is called the **name** of a . We regard M itself as a structure of L_M by setting the interpretation of i_a to be a for each $a \in M$.

Let L be a first-order language with universe M .

We define a **valuation** function $v = v_M$ from the set of closed formulas of L_M to the set $\{T, F\}$ of truth values inductively on the rank of sentences:

- $v(pt_1 \cdots t_n) := T$ if $p_M((t_1)_M, \dots, (t_n)_M)$ (equivalently if $((t_1)_M, \dots, (t_n)_M) \in p_M$) for n -ary relation symbols p and closed terms t_1, \dots, t_n ,

$$v(pt_1 \cdots t_n) := \begin{cases} T & \text{if } p_M((t_1)_M, \dots, (t_n)_M) \\ F & \text{otherwise} \end{cases}$$

- $v(\neg A) := T$ if $v(A) = F$,

$$v(\neg A) := \begin{cases} T & \text{if } v(A) = F \\ F & \text{otherwise} \end{cases}$$

- $v(A \vee B) := T$ if $v(A) = T$ or $v(B) = T$,

$$v(A \vee B) := \begin{cases} T & \text{if } v(A) = T \text{ or } v(B) = T \\ F & \text{otherwise} \end{cases}$$

- $v(\exists x A) := T$ if $v(A[i_a/x]) = T$ for some $a \in M$.

$$v(\exists x A) := \begin{cases} T & \text{if } v(A[i_a/x]) = T \text{ for some } a \in M \\ F & \text{otherwise} \end{cases}$$

If A is any formula of L , its **closure** is given by $\bar{A} := \forall x_1 \cdots \forall x_n A$ where x_1, \dots, x_n are the free variables of A . Define $v(A) := v(\bar{A})$.

If $v(A) = T$, we say A is **valid** in M and write $M \models A$ (“ M satisfies A ”). If A is not valid in M , write $M \not\models A$. Note that $M \models \neg A$ iff $M \not\models A$ and $M \models A \vee B$ iff $M \models A$ or $M \models B$.

For clarity, the following are equivalent by definition:

- (i) $v_M(A) = T$,
- (ii) A is valid in M ,
- (iii) A is true in M ,
- (iv) $M \models A$,
- (v) M satisfies A .

Similarly, the following are equivalent:

- (i) A is not valid in M ,
- (ii) A is false in M
- (iii) $v_M(A) = F$,
- (iv) M does not satisfy A ,
- (v) $M \not\models A$.
- (vi) $M \models \neg A$.

Exercise.

- (a) $M \models A \wedge B$ iff $M \models A$ and $M \models B$.
- (b) $M \models \forall v \varphi[v]$ iff $M \models \varphi[i_a/v]$ for all $a \in M$.
- (c) $M \models A \rightarrow B$ iff $M \not\models A$ or $M \models B$.
- (d) $M \models A \leftrightarrow B$ iff both A and B are valid or both not valid in the structure M .

Exercise. Let $A[v_1, \dots, v_n]$ be a formula and t_1, \dots, t_n be closed (i.e., variable-free) terms of L . Then

$$\forall v_1 \dots \forall v_n A \rightarrow A[t_1, \dots, t_n]$$

and

$$A[t_1, \dots, t_n] \rightarrow \exists v_1 \dots \exists v_n A$$

are valid in all structures of L .

Let T be a first-order theory.

A **model** of T is a structure of $L(T)$ in which all nonlogical axioms of T are valid.

Let T be a first-order theory and A a formula of T .

If A is valid in all models of T , say A is **valid** in T and write $T \models A$.

If A is not valid in some model of T , write $T \not\models A$.

Let M be a structure of L .

Define $\text{Th}(M)$ to be the set of all sentences of L that are true in M , called the **theory of M** .

A class \mathcal{M} of structures of a language L is called **elementary** if there is a theory T with language L such that the elements of \mathcal{M} are exactly the models of T .

model theory stuff. morphisms and so on as you would expect.

Let M be a structure of a language L .

A subset $X \leq M^n$ is called **definable** in L if there is a formula $\varphi[x_1, \dots, x_n, y_1, \dots, y_m]$ of L and **parameters** $\bar{b} \in M^m$ such that

$$\bar{a} \in X \quad \text{iff} \quad M \models \varphi[i_{\bar{a}}, i_{\bar{b}}].$$

If $\bar{b} \in A^n$ for some $A \subseteq M$, say X is **A -definable**.

A function $f : M^k \rightarrow M^\ell$ is **definable** if its graph is definable.

An element $a \in M$ is **A -definable** if the singleton $\{a\}$ is A -definable.

The **language of a propositional logic** L consists of the following data:

- (i) **variables**: a nonempty set of symbols,
- (ii) **logical connectives**: \neg (negation) and \vee (disjunction),

The set of **formulas** \mathcal{F} is the smallest set of expressions in L that is closed under negation $\neg A$ and disjunction $A \vee B$.

Let L be the language of a propositional logic. A **truth valuation** or **interpretation** or **structure** of L is a map v from the set of all variables of L to the set $\{T, F\}$ of truth values.

There is a natural extension of v to \mathcal{F} :

- $v(\neg A) = T$ iff $v(A) = F$,
- $v(A \vee B) = T$ iff $v(A) = T$ or $v(B) = T$.

If $v(A) = T$, say A is **true** in the structure v or that v **satisfies** A ; otherwise A is **false** in the structure v .

Let $\mathcal{A} \subseteq \mathcal{F}$ be a set of formulas in a propositional logical language L .

An interpretation v is called a **model** of \mathcal{A} if every $A \in \mathcal{A}$ is true in v . In which case, write $v \models \mathcal{A}$.

That is, $v \models \mathcal{A}$ whenever $v(A) = T$ for all $A \in \mathcal{A}$.

If \mathcal{A} has a model, say \mathcal{A} is **satisfiable**.

Let A and B be formulas of L .

Say A is a **tautological consequence** of \mathcal{A} , written $\mathcal{A} \models A$, if A is true in every model of \mathcal{A} .

If A is a tautological consequence of the empty set, say A is a **tautology** and write $\models A$. So A is a tautology iff $v(A) = T$ for all structures v of L .

If $A \leftrightarrow B$ is a tautology (i.e., $v(A) = v(B)$ for all v), say A and B are **tautologically equivalent** and write $A \equiv B$.

Exercise. $\mathbb{B} = \mathcal{F}/\equiv$ is a Boolean algebra on equivalence classes of formulas $[A]$ with the following operations:

$$\begin{aligned} 0 &= [A \wedge \neg A], \\ 1 &= [A \vee \neg A], \\ [A]' &= [\neg A], \\ [A] \vee [B] &= [A \vee B], \\ [A] \wedge [B] &= [A \wedge B]. \end{aligned}$$

Proposition. Let A, A_1, A_2, \dots, A_n be formulas of a propositional logic language. Then the following are equivalent:

- (a) A is a tautological consequence of A_1, \dots, A_n (i.e., $\{A_1, \dots, A_n\} \models A$),
- (b) $A_1 \rightarrow A_2 \rightarrow \dots \rightarrow A_n \rightarrow A$ is a tautology.

A formula A is said to be in **conjunctive normal form** (CNF) if it is written as

$$A = \bigwedge_{i=1}^k \bigvee_{j=1}^{n_i} A_{ij},$$

where each A_{ij} is a variable or its negation. That is, A is a conjunction of disjunctions of literals.

A formula A is said to be in **disjunctive normal form** (DNF) if it is written as

$$A = \bigvee_{i=1}^k \bigwedge_{j=1}^{n_i} A_{ij},$$

where each A_{ij} is a variable or its negation. That is, A is a disjunction of conjunctions of literals.

Fact. Every formula of a propositional logic language is tautologically equivalent to a formula in CNF and a formula in DNF.

Intuition. Recall that tautologies in propositional logic are those formulas that are true in every interpretation. For any propositional statement, we can find out if it is a tautology by constructing a truth table, i.e., by systematically examining all possible truth valuations of the relevant variables. For short propositions this is fine, and even for longer propositions it isn't hard to get a computer to crank through the possibilities. However, looking ahead to more complex logics, this approach quickly becomes infeasible. What we want is a structured way to find out if a formula is a tautology without having to check every possible interpretation. More generally, we want to check if a formula is a tautological consequence of a set of formulas. This is one way of looking at a problem like "if we assume X , Y , and Z , does it follow that A ?"

This is the motivation behind proofs. A proof should be a way of deducing tautological consequences without having to check every possible interpretation. In particular, we want to be able to shuffle around the statements of core facts pertaining to X , Y , and Z and show how they work together to necessitate A , disregarding exactly what the facts pertain to.

Henkin Extension The *Henkin Extension* T' of a first-order theory T is conservative.

Proof. We assume without loss of generality that T' is the extension of T by a (witness) constant c and the nonlogical axiom $\psi_c = \exists x F \rightarrow F[c/x]$, as T' is simply the limit of such of such extensions. (needs better justification, but only relies on fact that proofs are finite.)

(assume we break down proofs into only axioms and logical inference rules, i.e., extract theorems from T and inline proofs)

We must show that for any formula A of T which is provable in T' ($T' \vdash A$), there is also a proof in T ($T \vdash A$). Suppose A_1, \dots, A_n is a proof of $A = A_n$ in T' , where A is any formula of T' (not necessarily in A), i.e.,

$$T', A_1, \dots, A_{k-1} \vdash A_k,$$

for each $k < n$.

We will prove by (strong) induction on n that $T \vdash \exists t A_k[t/c]$ for all $k = 1, \dots, n$, where t is a variable that does not appear in any of A_1, \dots, A_n .

In the base case, $n = 0$, we have $T' \vdash A$, which requires that A is an axiom of T' (since we fully explicate all our proofs). By definition of T' , either A is an axiom of T (in which case $T \vdash A$ trivially) or $A = \psi_c$. In the latter case, we have

$$\begin{aligned} T \vdash \exists x F \rightarrow \exists t F[t/x] & \quad \text{tautology} \\ \sim \exists t (\exists x F \rightarrow F[t/x]) & \quad \text{\textcolor{red}{}\exists-\vee-Distributivity}} \\ = \exists t (\exists x F \rightarrow F[c/x])[t/c] & \\ = \exists t A[t/c]. & \end{aligned}$$

This completes the base case of the induction.

For the inductive step, assume that $T \vdash \exists t A_k[t/c]$ for all $k < n$. Now A is either an axiom of T' —in which case we are done by the same method as the base case—or A is the result of applying a logical inference rule to some A_k 's.

There are five cases to consider: (i) expansion, (ii) contraction, (iii) associative, (iv) cut, and (v) \exists -introduction.

(i) Expansion: $A = B \vee A_k$ for some $k < n$.

$$\begin{aligned} T \vdash \exists t A_k[t/c] & \quad \text{inductive hypothesis} \\ \vdash \exists t B[t/c] \vee \exists t A_k[t/c] & \quad \text{expansion rule} \\ = (\exists t B \vee \exists t A_k)[t/c] & \\ \sim \exists t (B \vee A_k)[t/c] & \quad \text{\textcolor{red}{}\exists-\vee-Distributivity}} \\ = \exists t A[t/c] & \end{aligned}$$

(ii) Contraction: $A_k = A \vee A$ for some $k < n$.

$$\begin{array}{ll}
T \vdash \exists t A_k[t/c] & \text{inductive hypothesis} \\
= \exists t (A \vee A)[t/c] & \\
\sim \exists t A[t/c] \vee \exists t A[t/c] & \text{\textcolor{red}{}\exists\text{-}\vee\text{-Distributivity}} \\
\vdash \exists t A[t/c] & \text{Contraction Rule}
\end{array}$$

(iii) Associative: $A = (B \vee C) \vee D$ with $A_k = B \vee (C \vee D)$ for some $k < n$.

$$\begin{array}{ll}
T \vdash \exists t A_k[t/c] & \text{Inductive Hypothesis} \\
= \exists t (B \vee (C \vee D))[t/c] & \\
\sim \exists t B[t/c] \vee (\exists t C[t/c] \vee \exists t D[t/c]) & \text{\textcolor{red}{}\exists\text{-}\vee\text{-Distributivity}} \\
\vdash (\exists t B[t/c] \vee \exists t C[t/c]) \vee \exists t D[t/c] & \text{Associativity Rule} \\
\sim \exists t ((B \vee C) \vee D)[t/c] & \text{\textcolor{red}{}\exists\text{-}\vee\text{-Distributivity}} \\
= \exists t A[t/c]. &
\end{array}$$

(iv) Cut: $A = C \vee D$, $A_k = B \vee C$ and $A_\ell = \neg B \vee D$ for some $k, \ell < n$.

$$\begin{array}{ll}
T \vdash \exists t A_k[t/c] & \text{Inductive Hypothesis} \\
= \exists t (B \vee C)[t/c] & \\
\sim \exists t B[t/c] \vee \exists t C[t/c] & \text{\textcolor{red}{}\exists\text{-}\vee\text{-Distributivity}}
\end{array}$$

and

$$\begin{array}{ll}
T \vdash \exists t A_\ell[t/c] & \text{Inductive Hypothesis} \\
= \exists t (\neg B \vee D)[t/c] & \\
\sim \exists t (\neg B)[t/c] \vee \exists t D[t/c] & \text{\textcolor{red}{}\exists\text{-}\vee\text{-Distributivity}}
\end{array}$$

(now need something that gives

$$(\exists t X \vee Y) \wedge (\exists t \neg X \vee Z) \vdash Y \vee Z.$$

but there are problems here. can't be what we want, since you can absolutely have a t satisfying X and another one not satisfying it and have both Y and Z be false. so need something stronger or that deals with \exists better.)

(v) \exists -Introduction: $A = \exists x B \rightarrow C$ and $A_k = B \rightarrow C$, where x is not free in C and $k < n$.

$$\begin{array}{ll}
T \vdash \exists t A_k[t/c] & \text{Inductive Hypothesis} \\
= \exists t (B \rightarrow C)[t/c] & \\
\sim \exists t B[t/c] \rightarrow \exists t C[t/c] & \text{\textcolor{red}{}\exists\text{-}\rightarrow\text{-Distributivity}} \\
\vdash \exists x \exists t B[t/c] \rightarrow \exists t C[t/c] & \text{\textcolor{red}{}\exists\text{-Introduction Rule}} \\
\sim \exists x \exists t B[t/c] \rightarrow \exists t C[t/c] & \text{\textcolor{red}{}\exists\text{-commutativity/interchange}}
\end{array}$$

