

Mathematical Logic

1 Propositional Logic

Before talking about propositional logic, let us review the basic connectives.

Name	Notation	Meaning
Negation	\neg	not
Conjunction	\wedge	and
Disjunction	\vee	or
Implication	\rightarrow	if, ..., then
Equivalence	\equiv	if and only if

Definition 1.1 (Set of atomic propositions). Given an arbitrary finite or countable set $AP = \{p_1, p_2, \dots\}$ which we call the set of atomic propositions. The elements in the set describe our propositions.

We will later see why this set is defined to be finite or countable.

Definition 1.2 (Propositions of depth n). Given a set AP of atomic propositions, write $P_0 = AP$, and call the set P_0 the set of propositions of depth 0. For any $n \geq 1$ define recursively the set P_n as the set of any element A of one of the following forms:

- (1) $A = \neg B$ for some $B \in P_{n-1}$.
- (2) $A = B * C$ for some propositions B and C of depth less than n , with at least one of them of depth $n-1$ and some connective $*$ $\in \{\wedge, \vee, \rightarrow, \equiv\}$.

The set P_n is called the set of propositions of depth n .

Definition 1.3 (Set of propositions). Given a set AP of atomic propositions, the set of all propositions is denoted P as defined as the unions of all propositions of all depths:

$$P := \bigcup_{n=0}^{\infty} P_n.$$

Remark 1.1. Notice that by our construction, there does not exist a proposition with an infinite number of atomic propositions. Thus, the set of atomic proposition needs to contain at most a countable amount of atomic propositions, and that is why we defined it to be countable in the first place.

Definition 1.4 (Assignment). An assignment is a function $M: AP \rightarrow \{T, F\}$.

Let A be a proposition in P . We say that a certain assignment M models P if the truth value of P under that assignment is T , and we denote $M \models P$. A rigorous definition for when a truth value of a proposition is T can be defined by using structural induction.

Definition 1.5 (Tautology). A proposition $A \in P$ is called a tautology if for any assignment M we have $M \models A$.

Definition 1.6 (Contradiction). A proposition $A \in P$ is called a contradiction if for any assignment M we have $M \not\models A$.

Definition 1.7 (Logical equivalence). We say that two propositions $A, B \in P$ are logically equivalent if for any assignment M we either have $M \models A$ and $M \models B$ or $M \not\models A$ and $M \not\models B$.

Remark 1.2. Logical equivalence is an equivalence relation.

Proposition 1.1. For every proposition $A \in P$, the propositions A and $\neg(\neg A)$ are logically equivalent.

Proof. Let M be an assignment such that $M \models A$. Then by definition we have that $M \not\models \neg A$. Finally, by definition we have that $M \models \neg(\neg A)$.

Otherwise we have $M \not\models A$. Then by definition we have that $M \models \neg A$. Finally, by definition we have that $M \not\models \neg(\neg A)$. \square

Definition 1.8 (Opposite assignment). For each assignment $M: AP \rightarrow \{T, F\}$ define $\overline{M}: AP \rightarrow \{T, F\}$ as:

$$\overline{M}(p) = \begin{cases} T, & M(p) = F \\ F, & M(p) = T \end{cases}.$$

The assignment \overline{M} is called the opposite assignment of M .

Informally, for any proposition P we define \overline{P} the proposition such that any appearance of $p \in AP$ in P is replaced by $\neg p$.

Proposition 1.2. For any assignment M and proposition P we have

$$M \models P \iff \overline{M} \models \overline{P}.$$

Definition 1.9 (n -ary truth function). An n -ary truth function is a function $f: \{T, F\}^n \rightarrow \{T, F\}$.

Definition 1.10 (Truth function representation). Let A be a proposition comprised of n atomic variables $\{p_i\}_{i=1}^n$, and let f be an n -ary truth function. We say that A represents f if for every assignment M we have

$$M \models A \iff f(M(p_1), \dots, M(p_n)) = T.$$

Example 1.1. The proposition $A = p_1 \wedge \neg p_2$ represents the following truth function:

$$f(x, y) = T \iff x = T \text{ and } y = F.$$

Definition 1.11 (Complete set of connectives). A set of connectives K is called complete if any n -ary truth function can be represented using propositions that use connectives from K alone.

Proposition 1.3. The following sets of connectives are complete:

$$(1) \{\neg, \wedge, \vee\}.$$

$$(2) \{\neg, \vee\}.$$

$$(3) \{\neg, \wedge\}.$$

$$(4) \{\neg, \rightarrow\}.$$

Definition 1.12 (Literal). A literal is an atomic formula or its negation.

Definition 1.13 (Disjunctive normal form). A logical formula is considered to be in DNF if it is a disjunction of one or more conjunctions of one or more literals.

Remark 1.3. A DNF formula is in full disjunctive normal form if each of its variables appears exactly once in every conjunction and each conjunction appears at most once (up to the order of variables).

Proposition 1.4. For every natural n and for any n -ary truth function f , there exists a proposition A in DNF form representing f .

Proposition 1.5. The set $\{\neg\}$ is not complete.

Proposition 1.6. *The set $\{\wedge, \vee\}$ is not complete.*

Definition 1.14 (Monotonic truth function). An n -ary truth function is called monotonic if there exists $\alpha \in \{T, F\}^n$ such that $f(\alpha) = T$ and $f(\beta) = T$ if and only if for every $1 \leq i \leq n$ such that $\alpha_i = T$ we have $\beta_i = T$.

Remark 1.4. The last definition is intuitive because we can change the order of coordinates of α such that all the T coordinates are an initial segment of α , and thus $f(\beta) = T$ if in that binary sense $\beta > \alpha$.

Proposition 1.7. *A n -ary truth function can be represented using propositions that use connectives from $\{\wedge, \vee\}$ alone if and only if it is monotonic.*

From now on we will assume all propositions are written using only the connectives $\{\neg, \rightarrow\}$.

Definition 1.15 (Propositional proof system). A propositional proof system is comprised of

- (1) A set of propositions called axioms.
- (2) A set of rules of inference that allow us to deduce propositions from previous propositions.

Definition 1.16 (Proof). Given a set of propositions Σ which we call the hypotheses, and a proposition A , we say that A is provable from Σ , and denote $\Sigma \vdash A$ if there exists a finite sequence of propositions B_1, \dots, B_n such that $B_n = A$ and for each $1 \leq i \leq n$, one of the following holds:

- B_i is an axiom.
- $B_i \in \Sigma$ (B_i is a hypothesis).
- B_i is deducible from the propositions preceding it in the sequence according to the rules of inference.

We call the sequence $\{B_i\}_{i=1}^n$ a proof for A .

Remark 1.5. In the case $\Sigma = \emptyset$ we denote $\vdash A$.

The following standard proof system is the standard proof system we use, and is sometimes called a Hilbert system.

Definition 1.17 (Standard propositional proof system). The standard propositional proof system is comprised of the following three axiom schemes:

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

Where p, q, r are propositions, and the following rule of inference

$$\frac{p \quad p \rightarrow q}{\therefore q}$$

which is called Modus Ponens (MP).

Remark 1.6. Notice that we wrote three axioms schemes in the system, but we actually have infinitely many axioms.

Remark 1.7. All axioms are tautologies.

Proposition 1.8. *For every proposition A we have $\vdash A \rightarrow A$.*

Definition 1.18 (Weak soundness). A propositional system is called sound if every provable proposition is a tautology. This can be written as

$$\vdash A \Rightarrow \models A.$$

Definition 1.19 (Weak completeness). A propositional system is called complete if every tautology is provable. This can be written as

$$\models A \Rightarrow \vdash A.$$

Theorem 1.9. (Weak Soundness Theorem). *The standard proof system is sound.*

Proof. To be added. □

Theorem 1.10. (Weak Completeness Theorem). *The standard proof system is complete.*

In order to prove the completeness theorem we first have to prove some other theorems.

Theorem 1.11. (Propositional Deduction Theorem). *Let Σ be a set of propositions, let A and B be propositions. Then the following are equivalent*

- (1) $\Sigma \cup \{A\} \vdash B$.
- (2) $\Sigma \vdash A \rightarrow B$.

Proof. To be added. □

Remark 1.8. By the deduction theorem, it suffices to prove that $A \vdash A$ (which is trivial) to prove that $\vdash A \rightarrow A$ which was less trivial.

Proposition 1.12. *Let A and B be propositions.*

- (1) $\vdash \neg A \rightarrow (A \rightarrow B)$.
- (2) $\vdash (\neg A \rightarrow A) \rightarrow A$.
- (3) $\vdash A \rightarrow \neg\neg A$.
- (4) $\vdash (A \rightarrow B) \rightarrow ((\neg A \rightarrow B) \rightarrow B)$.

Proposition 1.13 (Principle of explosion). *Let A and B be propositions. Then*

$$A \rightarrow (\neg A \rightarrow B).$$

Definition 1.20 (Consistency). A set of propositions Σ is called consistent if there does not exist a proposition A such that $\Sigma \vdash A$ and $\Sigma \vdash \neg A$. If Σ is not consistent it is said to be inconsistent.

Proposition 1.14. *Let Σ be a set of propositions. Then the following are equivalent:*

- (1) Σ is inconsistent.
- (2) every proposition A is provable from Σ .

Definition 1.21 (Completeness of a set of propositions). A set of propositions Σ is said to be complete for every proposition A we have

$$\Sigma \vdash A \quad \text{or} \quad \Sigma \vdash \neg A.$$

We can conclude that if Σ is consistent and complete then for every proposition A exactly one of the following is true:

- (1) $\Sigma \vdash A$.
- (2) $\Sigma \vdash \neg A$.

Theorem 1.15. (The expansion theorem). *Let Σ be a consistent set of propositions. Then there exists a consistent set of propositions Σ' such that $\Sigma \subset \Sigma'$.*

Lemma 1.16. *Let A be a proposition, let Σ be a set of propositions such that $\Sigma \not\vdash A$. Then $\Sigma \cup \{\neg A\}$ is consistent.*

Remark 1.9. Notice that if Σ were inconsistent, then from the principle of inclusion and the deduction theorem we get that $\Sigma \vdash A$. That is the reason why Σ is not required to be consistent.

Proof. Two proofs needed here. □

Lemma 1.17. *Let Σ be a complete and consistent set of propositions. Then there exists an assignment V such that for any proposition A we have*

$$\Sigma \vdash A \iff V \models A$$

Proof. Define the assignment $A: AP \rightarrow \{T, F\}$ as such:

$$V(p) := \begin{cases} T, & \Sigma \vdash p \\ F, & \Sigma \vdash \neg p \end{cases}.$$

It is well defined since Σ is complete and consistent. The proof this is the correct assignment follows from structural induction. □

We can now proceed to prove Theorem 1.10.

Proof. To be added. □

Definition 1.22 (Semantic consequence). Let Σ be a set of propositions, let A be a proposition. We denote $\Sigma \models A$ if for each assignment V that satisfies each $P \in \Sigma$ we have $V \models A$. We also say that A is a semantic consequence of Σ .

Remark 1.10. Notice that in particular $\emptyset \models A$ if and only if A is a tautology.

Theorem 1.18. (Strong Soundness Theorem). *Let Σ be a set of propositions, let A be a proposition. Then*

$$\Sigma \vdash A \implies \Sigma \models A.$$

Theorem 1.19. (Strong Completeness Theorem). *Let Σ be a set of propositions, let A be a proposition. Then*

$$\Sigma \vdash A \iff \Sigma \models A.$$

Definition 1.23 (Model). Let Σ be a set of propositions, let M be an assignment. Then M is said to be a model of Σ if $M \models A$ for all $A \in \Sigma$.

The following is a corollary from Theorem 1.18 and Theorem 1.19.

Corollary 1.20. *Let Σ be a set of propositions. Then Σ is consistent if and only if there exists a model to it.*

Proof. (\Rightarrow) Assume by contradiction that Σ has no model. Then by definition we have vacuously that $\Sigma \models B$ for any proposition B . Then by the strong completeness theorem we have that $\Sigma \vdash B$ for any B . In particular for an arbitrary proposition A we have $\Sigma \vdash A$ and $\Sigma \vdash \neg A$ which shows that Σ is inconsistent. This contradiction completes the proof of this direction.

(\Leftarrow) Assume by contradiction that Σ is inconsistent. Then there exists some proposition A such that $\Sigma \vdash A$ and $\Sigma \vdash \neg A$. Then by the strong soundness theorem we have that $\Sigma \models A$ and $\Sigma \models \neg A$. Since there does not exist an assignment V such that $V \models A$ and $V \models \neg A$ we have that there does not exist a model for Σ . \square

Theorem 1.21. (Compactness theorem). *Let Σ be a set of propositions. Then there exists a model for Σ if and only if there exists a model for any finite subset of Σ .*

Proof. (\Rightarrow) Let M be a model for Σ . Then in particular M is a model for any finite subset of Σ .

(\Leftarrow) Now suppose that there does not exist a model for Σ . From the above corollary that means that Σ is inconsistent. Thus exists a proposition A such that $\Sigma \vdash A$ and $\Sigma \vdash \neg A$. Let $\Sigma' \subset \Sigma$ be the set of all propositions from the proofs of A and $\neg A$. Then Σ' is finite since all proofs are finite by definition, and Σ' is inconsistent because it can prove both A and $\neg A$. Thus, from the corollary above, there exists no model to Σ' . \square

Theorem 1.22. (De Bruijn–Erdős theorem). *Let $G = (V, E)$ be a countable graph. Then the following are equivalent:*

- (1) G is k -colorable.
- (2) every finite subgraph of G is k -colorable.

Corollary 1.23. *Every planar graph is 4-colorable.*

Talk about Ramsey + Find the Ramsey theorem tex.

Definition 1.24. Let Σ be a set of propositions. Define

$$\mu(\Sigma) := \{M \mid M \models \Sigma\}.$$

Definition 1.25 (Definability). A set of assignments \mathcal{M} is said to be definable if there exists a proposition set Σ such that $\mu(\Sigma) = \mathcal{M}$. If \mathcal{M} is definable by a finite proposition set Σ it is said to be finitely definable.

Example 1.2. The set of all assignments $\mathcal{M} = \{F, T\}^n$ is definable by any set of tautologies since all assignments satisfy all tautologies.

Example 1.3. The set $\mathcal{M} = \emptyset$ is definable by the proposition $A = p \rightarrow \neg p$ since there does not exist an assignment that satisfies A .

Proposition 1.24. *Most sets of assignments are undefinable.*

Proof. It is clear that the cardinality of the set of all definable sets of assignments is smaller than the cardinality of the set of all sets of proposition. Therefore there are at most 2^{\aleph_0} propositions.

Since the cardinality of the set of sets of assignments is $2^{2^{\aleph_0}}$ the proposition follows. \square

Proposition 1.25. *For every assignment V the set $\{V\}$ is definable.*

Proof. Define

$$\Sigma := \{p \in AP \mid V(p) = T\} \cup \{\neg p \in AP \mid V(p) = F\}.$$

We have that $\mu(\Sigma) = \{V\}$ which completes the proof. \square

Proposition 1.26. *The set $\mathcal{M} = \{M \mid M \text{ is monotonic}\}$ is definable. In this question M is monotonic if for all $i < j$ we have $M(p_i) \Rightarrow M(p_j)$.*

Proposition 1.27 (Properties of definable sets). (1) if $\Sigma_1 \subset \Sigma_2$ then $\mu(\Sigma_1) \supset \mu(\Sigma_2)$.

(2) let $\{\mathcal{M}_i\}_{i \in I}$ be a set of definable sets. Then $\cap_{i \in I} \mathcal{M}_i$ is definable.

(3) a finite union of definable sets is definable.

(4) any finitely definable set is definable by a single proposition.

Proof. (1) trivial.

(2) for any \mathcal{M}_i there exists a set of propositions Σ_i such that $\mu(\Sigma_i) = \mathcal{M}_i$. We will show that

$$\bigcap_{i \in I} \mathcal{M}_i = \mu\left(\bigcup_{i \in I} \Sigma_i\right).$$

We have that

$$V \in \mu\left(\bigcup_{i \in I} \Sigma_i\right) \iff \forall i \in I: V \models \Sigma_i \iff \forall i \in I: V \in \mu(\Sigma_i) = \mathcal{M}_i \iff V \in \bigcap_{i \in I} \mathcal{M}_i.$$

(3) it suffices to prove that if \mathcal{M}_1 and \mathcal{M}_2 are definable then $\mathcal{M}_1 \cup \mathcal{M}_2$ is definable. Define

$$\Sigma := \{A_1 \vee A_2 \mid A_1 \in \Sigma_1 \wedge A_2 \in \Sigma_2\}.$$

We will show that $\mu(\Sigma) = \mathcal{M}_1 \cup \mathcal{M}_2$. It is clear that $\mathcal{M}_1 \cup \mathcal{M}_2 \subset \mu(\Sigma)$. Let $V \in \mu(\Sigma)$, and assume by contradiction that $V \notin \mathcal{M}_1 \cup \mathcal{M}_2$. Then there exist $A \in \Sigma_1$ and $B \in \Sigma_2$ such that $V \not\models A$ and $V \not\models B$. Since we have $A \vee B \in \Sigma$ we get that $V \notin \mu(\Sigma)$. This contradiction shows that $\mu(\Sigma) \subset \mathcal{M}_1 \cup \mathcal{M}_2$ which completes the proof.

(4) let \mathcal{M} be finitely definable, let $\mathcal{M} = \mu(\Sigma)$ such that Σ is finite. Define

$$\Sigma' := \left\{ \bigwedge_{i=1}^n A_i \mid A_i \in \Sigma \right\}.$$

It is clear that $\mu(\Sigma') = \mathcal{M}$ which completes the proof. □

Consider the space of all assignment $S := \{F, T\}^{AP}$.

Definition 1.26 (Convergence of assignments). Let $\{M_i\}_{i=1}^\infty \subset S$. We say that $\{M_i\}_{i=1}^\infty$ converges if for every $p \in AP$ the sequence $\{M_n(p)\}_{n=1}^\infty$ stabilizes eventually. If $\{M_i\}_{i=1}^\infty$ converges, we say that its limit is the assignment M such that $M(p)$ is the value to which $\{M_n(p)\}_{n=1}^\infty$ stabilizes.

Definition 1.27 (Closed set of assignments). Let \mathcal{M} be a set of assignments. We say that \mathcal{M} is closed if the limit of any convergent sequence $\{M_n\}_{n=1}^\infty \subset \mathcal{M}$ is in \mathcal{M} .

Example 1.4. The set $\mathcal{M} = \{M \mid M \text{ is monotonic}\} \setminus \{m_0\}$ such that $m_0(p) = T$ for any $p \in AP$ is not closed. Choose the sequence $\{M_n\}_{n=1}^\infty$ such that

$$M_n := \begin{cases} T, & i \leq n \\ F, & i > n \end{cases}.$$

The sequence $\{M_n\}_{n=1}^\infty$ converges to m_0 but $m_0 \notin \mathcal{M}$.

Proposition 1.28. *Let \mathcal{M} be a set of assignment. Then*

$$\mathcal{M} \text{ is closed} \iff \mathcal{M} \text{ is definable.}$$

Proof. To be added. □

Proposition 1.29. *Let \mathcal{M} be a set of assignments. Then the following are equivalent:*

- (1) \mathcal{M} is finitely definable.
- (2) \mathcal{M} is definable by a single proposition.
- (3) \mathcal{M} is definable and \mathcal{M}^c is definable.

Proof. We proved direction (1) \rightarrow (2) before.

(2) \rightarrow (3) let $\mathcal{M} = \mu(\{A\})$. Then $\mathcal{M} = \mu(\{\neg A\})$.

(3) \rightarrow (1) to be added. □

2 Structures

Definition 2.1 (Structure). A structure \mathcal{M} is composed of

- A nonempty set $U^{\mathcal{M}}$ called the universe of \mathcal{M} ;
- Constants, which are distinguished elements of $U^{\mathcal{M}}$;
- Functions of the form $f: (U^{\mathcal{M}})^n \rightarrow U^{\mathcal{M}}$;
- Relations of the form $R \subseteq (U^{\mathcal{M}})^n$.

Example 2.1. The structure of the integers as an additive group would be defined as such:

- $U^{\mathcal{M}} = \mathbb{Z}$;
- $e^{\mathcal{M}} = 0$ as the zero element (a constant);
- $f^{\mathcal{M}}$ a binary function defined as $f(z_1, z_2) = z_1 + z_2$ (addition function);
- $g^{\mathcal{M}}$ a unary function defined as $g(z) = -z$ (inverse function);
- No relations except the equality relation.

Definition 2.2 (Language). A Language \mathcal{L} is composed of

- Constant symbols (c, d, \dots) ;
- Function symbols (f, g, \dots) , each one with a given arity;
- Relation symbols (R, P, \dots) , each one with a given arity;
- Binary equality symbol $=$ (optional);
- Connectives $(\wedge, \vee, \neg, \rightarrow, \equiv)$;
- Variables (x, y, z) ;
- Quantifiers (\exists, \forall) .

Definition 2.3 (A Language for a model). We say that \mathcal{L} is a model for a model \mathcal{M} if

- For each constant symbol $c \in \mathcal{L}$ there exists a constant $c^{\mathcal{M}} \in \mathcal{M}$.
- For each n -ary function symbol $f \in \mathcal{L}$ there exists a n -ary function $f^{\mathcal{M}} \in \mathcal{M}$.
- For each n -ary relation symbol $R \in \mathcal{L}$ there exists a n -ary relation $R^{\mathcal{M}} \in \mathcal{M}$.

Example 2.2 (Language of groups). The language of groups \mathcal{L}_G contains

- A single constant symbol called e (the unit);
- A single binary function symbol f (the group operation);
- A single unary function symbol g (the inverse operation);
- The equality symbol $=$ (for comparing elements).

We can see that the language \mathcal{L}_G is a language for the structure from Example 2.1.

Remark 2.1. Notice that the language \mathcal{L}_G is not specific for the model from Example 2.1, and it is a language for all models of a groups.

Definition 2.4 (Atomic term). An atomic term is a constant symbol, or a variable.

Definition 2.5 (Term). Denote T_0 the set of atomic terms. Recursively define the set T_n , assuming that the sets T_k for $k < n$ are all defined. Define

$$T_n := \left\{ f(x_1, \dots, x_k) \mid x_1, \dots, x_k \in \bigcup_{k < n} T_n \text{ and } f \text{ is a } k\text{-ary function symbol and } \exists i \text{ s.t. } x_i \in T_{n-1} \right\}$$

and then the set T defined as

$$T := \bigcup_{n=0}^{\infty} T_n$$

is called the set of terms, and a term is an element of T .

Let \mathcal{L} be a language. We can now define formulas in \mathcal{L} .

Definition 2.6 (Atomic formula). An atomic formula is an expression of the form $R(t_1, \dots, t_k)$ such that R is a k -ary relation symbol in \mathcal{L} and t_1, \dots, t_k are terms.

Definition 2.7 (Formula). Denote F_0 the set of atomic formulas. Recursively define the set F_n , assuming that the sets F_k for $k < n$ are all defined. An element of F_n (a formula of depth n) is one of the following:

- $\neg(\beta)$ such that $\beta \in F_{n-1}$;
- $\alpha * \beta$ for $*$ $\in \{\wedge, \vee, \rightarrow\}$, $\alpha, \beta \in \bigcup_{k < n} F_k$ and at least one of α, β is of depth $n-1$;
- $\exists x(\beta)$ such that $\beta \in F_{n-1}$ and x is a variable;
- $\forall x(\beta)$ such that $\beta \in F_{n-1}$ and x is a variable.

The set of formulas is defined as

$$F := \bigcup_{n=0}^{\infty} F_n$$

and a formula is an element of F .

Remark 2.2. The range in a formula where a quantifier is engaged in is called the *scope* of the quantifier.

Example 2.3 (Formulas of group theory). Recall the language of group theory \mathcal{L}_G . The formulas of group theory are

- $(\forall x)((f(x, e) = x) \wedge (f(e, x) = x))$ (Unit element);
- $(\forall x)((f(x, g(x)) = e) \wedge (f(g(x), x) = e))$ (Group operation is commutative);
- $(\forall x)(\forall y)(\forall z)(f(f(x, y), z) = f(x, f(y, z)))$ (Associativity);

Definition 2.8 (Substitution). Let \mathcal{M} be a model and let \mathcal{L} be a language for \mathcal{M} . Denote V the set of variables in \mathcal{L} . A substitution is a function $S: V \rightarrow W^{\mathcal{M}}$.

We can actually naturally expand a substitution S to a function of the form $S^*: T \rightarrow W^{\mathcal{M}}$. We define S^* inductively on the depth of the term $t \in T$.

- if t is a variable we define $S^*(t) = S(t)$;
- if t is a constant symbol we define $S^*(t) = c^{\mathcal{M}}$.

Now since we defined S^* on terms of depth 0 we can assume that S^* is defined on all terms of depth $k < n$ and define it for terms of depth n .

Let $t \in T_n$. Then t is of the form $f(t_1, \dots, t_k)$ for t_1, \dots, t_k for which the function S^* is already defined. We define

$$S^*(t) := f^{\mathcal{M}}(S^*(t_1), \dots, S^*(t_k)).$$

This completes the expansion we wanted.

Definition 2.9 (Satisfiability in a model). Let \mathcal{M} be a model and let \mathcal{L} be a language for \mathcal{M} . We will define when \mathcal{M} satisfies a formula α under a substitution S . In this case we denote $\mathcal{M} \models_S \alpha$.

In the case of formulas of depth 0 we have that $\alpha = R(t_1, \dots, t_k)$ for some $t_1, \dots, t_k \in T$. We define

$$\mathcal{M} \models_S \alpha \iff (S^*(t_1), \dots, S^*(t_k)) \in R^{\mathcal{M}}.$$

Inductively, for formulas of greater depths we define

- if $\alpha = \neg\beta$ then $\mathcal{M} \models_S \alpha$ if and only if $\mathcal{M} \not\models_S \beta$.
- if $\alpha = \beta \vee \gamma$ then $\mathcal{M} \models_S \alpha$ if and only if $\mathcal{M} \models_S \beta$ or $\mathcal{M} \models_S \gamma$.
- if $\alpha = \beta \wedge \gamma$ then $\mathcal{M} \models_S \alpha$ if and only if $\mathcal{M} \models_S \beta$ and $\mathcal{M} \models_S \gamma$.
- if $\alpha = \beta \rightarrow \gamma$ then $\mathcal{M} \models_S \alpha$ if and only if $\mathcal{M} \models_S \beta$ and $\mathcal{M} \models_S \gamma$.
- if $\alpha = \beta \equiv \gamma$ then $\mathcal{M} \models_S \alpha$ if and only if $\mathcal{M} \models_S \beta$ and $\mathcal{M} \models_S \gamma$ or $\mathcal{M} \not\models_S \beta$ and $\mathcal{M} \not\models_S \gamma$.
- if $\alpha = \exists x(\beta)$ then $\mathcal{M} \models_S \alpha$ if and only if exists a substitution S' that agrees with S on all variables except possibly for x we have $\mathcal{M} \models_{S'} \beta$.
- if $\alpha = \forall x(\beta)$ then $\mathcal{M} \models_S \alpha$ if and only if for all substitutions S' that agrees with S on all variables except possibly for x we have $\mathcal{M} \models_{S'} \beta$.

Definition 2.10 (Formula true in a model). Let \mathcal{M} be a model, let \mathcal{L} be a language for \mathcal{M} , and let α language \mathcal{L} . We say that α is true in \mathcal{M} if $\mathcal{M} \models_S \alpha$ for any substitution S . In this case, we write $\mathcal{M} \models \alpha$.

Definition 2.11 (Formula true in a language). Let \mathcal{M} be a model, let \mathcal{L} be a language for \mathcal{M} , and let α language \mathcal{L} . We say that α is true if $\mathcal{M} \models \alpha$ for all structures \mathcal{M} . In this case, we write $\models \alpha$.

Example 2.4. In the language of group theory, the formula

$$\alpha = (\forall x)(\forall y)(f(x, y) = f(y, x))$$

is true in all models of abelian groups, but is not true in the language of group theory in general.

Definition 2.12 (Bound and free instances of variables). For a formula α , and instance of a variable x in α is called bound if it is inside the scope of a quantifier, and it is called free otherwise.

Example 2.5.

$$\alpha = (\forall x)(\underbrace{R(x)}_{\text{bound instance}}) \wedge Q(\underbrace{x, y}_{\text{free instance}})$$

Definition 2.13 (Free variables). We say that a variable x is free in α if it has at least one free instance in α .

Remark 2.3. We denote by $FV(\alpha)$ the set of free variables in α .

Definition 2.14 (Sentence). Let α be a formula. If $FV(\alpha) = \emptyset$, then α is called a sentence.

Proposition 2.1. Let α be a formula. Suppose that S_1 and S_2 are substitutions such that $S_1(x) = S_2(x)$ for all $x \in FV(\alpha)$. Then

$$\mathcal{M} \models_{S_1} \alpha \iff \mathcal{M} \models_{S_2} \alpha.$$

Remark 2.4. We notice that for any model \mathcal{M} , substitution S , and formula β we have that

$$\mathcal{M} \models_S (\forall x)(\exists x)\beta \iff \mathcal{M} \models_S (\exists x)\beta$$

Corollary 2.2. If α is a sentence then $\mathcal{M} \models \alpha$ or $\mathcal{M} \models \neg\alpha$.

Remark 2.5. Let S be a substitution, let x be a variable, let $w \in W^{\mathcal{M}}$. From now on we denote $S^{x \leftarrow a}$ the substitution that agrees with S on all variables except for x , and $S^{x \leftarrow a}(x) = a$.

Definition 2.15 (Theory). A collection of sentences in predicate logic is called a theory.

Definition 2.16 (Structure of a theory). A structure \mathcal{M} is a structure in a theory Σ if for all $\alpha \in \Sigma$ we have $\mathcal{M} \models \alpha$.

Remark 2.6. We denote $\text{MOD}(\Sigma)$ to be the collection of all structures that satisfy Σ .

Remark 2.7. We denote $\text{Theory}(\mathcal{M})$ to be the collection of all formulas α such that $\mathcal{M} \models \alpha$.

Definition 2.17 (Elementary equivalence). We say that two structures $\mathcal{M}_1, \mathcal{M}_2$ are elementary equivalent if $\text{Theory}(\mathcal{M}_1) = \text{Theory}(\mathcal{M}_2)$.

Definition 2.18 (Isomorphism of structures). We say that two structures $\mathcal{M}_1, \mathcal{M}_2$ are isomorphic if exists a bijection $f: W^{\mathcal{M}_1} \rightarrow W^{\mathcal{M}_2}$ that preserves all functions and relations between the models and also for all constants $c^{\mathcal{M}_1}$ we have $f(c^{\mathcal{M}_1}) = c^{\mathcal{M}_2}$.

Proposition 2.3. Models that are isomorphic are elementarily equivalent.

Remark 2.8. Not all elementarily equivalent models are isomorphic.

Example 2.6 (Peano theory). The language \mathcal{L}_P has the constant symbol 0, the function symbols $+$ and $*$, the unary increment function symbol S , and the relation symbol $<$. The axioms (sentences) in the theory are:

- (1) $(\forall x)(\neg(S(x) = 0))$;
- (2) $(\forall x)(\forall y)(S(x) = S(y) \rightarrow x = y)$;
- (3) $(\forall x)(x + 0 = x)$;
- (4) $(\forall x)(\forall y)(x + S(y) = S(x + y))$;
- (5) $(\forall x)(x * 0 = 0)$;
- (6) $(\forall x)(\forall y)(S(x) * y = x * y + x)$;
- (7) $(\forall x)(\neg(x < 0))$;
- (8) $(\forall x)(\forall y)(x < S(y) \rightarrow (x < y) \vee (x = y))$;
- (9) $(\forall x)(\forall y)(x < y \vee y < x \vee x = y)$;
- (10) For every φ such that x is a free variable we have $\varphi_x(0) \rightarrow ((\forall x)(\varphi \rightarrow \varphi_x(S(x))) \rightarrow \varphi)$.

Similarly, we can define the theory of ZFC (set theory), field theory, group theory, and all other theories we have encountered so far.

Definition 2.19 (Free substitution). Given a formula α , a variable x , and a term t , a free substitution of t for x is a substitution of every free instance of x in α by t .

Definition 2.20 (Collision-free substitution). A free substitution of t for x in α is said to collide if there exists a variable y such that:

- (1) y appears in t ;
- (2) there exists a free instance of inside the scope of quantifiers of y like $\forall y$ or $\exists y$.

The variable y is said to be captured by α . If there are no collisions, the substitution is said to be collision-free.

Remark 2.9. We can see that a substitution is not collision-free if there exists a variable y in α such that after the substitution the number of bound instances of y increases.

Example 2.7. The substitution of x to y in α where α is defined as

$$\alpha := (\exists y)(\neg(x = \underbrace{y}_{\text{bound instance}}))$$

is not collision-free since y is captured by α . After the substitution we get

$$\alpha := (\exists y)(\neg(y = y))$$

which has two bound instances of y .