# Chapter 2

# Predicate Logic

Propositional logic is not able to describe all entailments which we consider to be "logically correct". This is because logical variables are the simplest formulas in propositional logic and they do not have any inner structure. And the simplest formulas do need inner structure in some entailments. Let us give an example of such deduction.

Consider the following inference:

$$\frac{\begin{array}{l}\text{Peter can play violin.}\\ \text{Everybody who can play violin has a musical ear.}\end{array}}{\text{Peter has a musical ear.}}$$

If we try to describe the above sentences as logical variables, then the first sentence would be one logical variable, say $a$. The second sentence resembles an implication of two formulas, say $b \Rightarrow c$. The third sentence is again a logical variable, say $d$. But $\{a, b \Rightarrow c\} \not\models d$.

We will see that the above inference is correct in predicate logic. Let us start with an informal description of formulas in predicate logic.

## 2.1 Syntactics of Predicate Logic

**2.1.1 An Informal Description of Predicate Logic.** For this purpose we will use predicates – properties, and quantifiers.

What do we intuitively need for a description of the sentence "Peter can play violin."? The statement speaks about "Peter" as an object and a property which Peter has, i.e. the property "to be able to play violin". Also the third sentence "Peter has a musical ear." has a similar structure. The property here is "to have a musical ear". The middle sentence speaks about "everybody", where by "everybody" we mean "every object" (every human being). In fact, it has a form of an implication: Every object that has the property "to be able to play violin", also has the property of "having a musical ear".

Let us denote by $V$ the property "to be able to play violin" and by $E$ the property "to have a musical ear". Now, we can write the above inference as:

$$\frac{\begin{array}{l}\text{Peter has } V.\\ \text{Everybody who has } V \text{, has } E.\end{array}}{\text{Peter has } E.}$$

Even this is a rather long description. We will write $V(p)$ instead of "Peter can $V$.", where $p$ denotes Peter. Similarly, we shorten the description of the third sentence, to $E(p)$. To shorten also the second sentence, we introduce an abbreviation for "everybody", in other words, "for all objects". We will denote it by $\forall$ and call it the *universal quantifier*. Any quantifier must be followed by a variable. The expression $\forall x$ is read as "for every $x$". Of course, we do not say only "for every" we must say "for every object" (and explain where are the objects from). Variables will be denoted by $x, y, \ldots$, and they represent objects. Now the second sentence has the following form: $\forall x\,(V(x) \Rightarrow E(x))$. Hence the whole entailment is written as follows:

$$\begin{array}{l} V(p) \\ \forall x\,(V(x) \Rightarrow E(x)) \\ \hline E(p) \end{array}$$

**2.1.2 Example.** We give another inference which is typical for predicate logic.

$$\begin{array}{l} \text{If a natural number is even, then its successor is odd.} \\ \text{The number 2 is even.} \\ \hline \text{The successor of the number 2 is odd.} \end{array}$$

Properties (predicate symbols) contained in the inference are: “to be an even number”, we will denote it by $E$, and “to be an odd number”, which will be denoted by $O$. Furthermore, we have one object (a constant symbol) which is the number 2. Finally, we need to describe “a successor of a natural number”. Here, by a successor we do not understand the property “to be a successor”, we do not ask whether a number is the successor of another one or not. Here we want to work with the successor of a natural number as with a natural number, i.e. with an object. Of course, this object is indirectly described — by means of a number which precedes it. Therefore, we will look at “a successor” as a function. We denote it by $f$, and it assigns to every natural number its successor, i.e. $f\colon n \mapsto n+1$. The symbol $f$ will be referred as a functional symbol.

Now, we are able to formalize the second and the third sentences: $E(2)$ and $O(f(2))$. The first sentence contains a quantifier. It is not completely clear from the English formulation which quantifier it should be. This ambiguity of the formulation, i.e. whether our statement should refer to “all” objects or only to “some” object, is common for nearly all natural languages. Hence we recommend to the reader to try to reformulate the sentence under consideration so that the quantification will be clearer. In our case, the first sentence has the same meaning as the statement “Whenever a natural number is even, its successor is an odd number.”. Therefore we get: $\forall x\,(E(x) \Rightarrow O(f(x)))$. The whole inference is:

$$\begin{array}{l} \forall x\,(E(x) \Rightarrow O(f(x))) \\ E(2) \\ \hline O(f(2)) \end{array}$$

## 2.2 Formal Description of Formulas in Predicate Logic.

Let us now give a formal definition of formulas of predicate logic. For this we need first to define a language of predicate logic which will capture ”objects”, ”properties”, ”special objects”, ”mappings”, and ”quantifiers”.

### 2.2.1 The Language of Predicate Logic.

**Definition.** A *language of predicate logic* consists of

1. *logical symbols*, i.e.:
   a) infinite countable set $\mathsf{Var}$ of individual variables
   b) propositional logical connectives: $\neg, \wedge, \vee, \Rightarrow, \Leftrightarrow$
   c) universal quantifier $\forall$ and existential quantifier $\exists$
2. *special symbols*, i.e.:
   a) a set $\mathsf{Pred}$ of predicate symbols (it cannot be empty)
   b) a set $\mathsf{Cons}$ of constant symbols (it may be empty)
   c) a set $\mathsf{Func}$ of functional symbols (it may be empty)
3. auxiliary symbols, such as brackets “[, ]”, parentheses “(, )”, and commas “,”.

□

For every predicate or functional symbol a natural number is given $n$, for a predicate symbol $n \geq 0$, for a functional symbol $n \geq 1$. This number indicates how many objects the predicate symbol concerns, or how many variables the functional symbol has. We call the number the *arity* of a predicate or functional symbol.

**Remark.** The fact that a predicate symbol $P$ has arity 0 means that "it does not concern any individual variable", so it is non structured and it can be considered as a logical variable. In such a way, propositional logic is contained in predicate logic.

**2.2.2 Terms.** First, we introduce a notion of a term: Roughly speaking, a term is an object which can be described using variables, constants and functional symbols also in a more complicated way. In the above examples the terms were for instance "Peter", "Peter's father", "a successor of the number 2". In the language of predicate logic, terms play the role of "substantives".

**Definition.** The set of *terms* is defined by the following rules:

1. Every variable and every constant symbol is a term.
2. If $f$ is a functional symbol of arity $n$ and $t_1, t_2, \ldots, t_n$ are terms, then $f(t_1, t_2, \ldots, t_n)$ is also a term.
3. Anything that was not created by a finite use of the above rules 1 and 2 is not a term.

□

**2.2.3 Atomic formulas.** If we know what are terms (i.e. objects with which our statements deal), we can start to form atomic formulas of predicate logic:

**Definition.** An *atomic formula* is a predicate symbol $P$ applied to as many terms as its arity is. In other words, if the arity of a predicate symbol $P$ is $n$ and $t_1, t_2, \ldots, t_n$ is an $n$-tuple of terms, then $P(t_1, t_2, \ldots, t_n)$ is an atomic formula. □

**2.2.4 Formulas of Predicate logic.** Analogously as in propositional logic we form also more complicated formulas than atomic ones.

**Definition.** The set of *formulas* is defined by the following rules:

1. Every atomic formula is a formula.
2. If $\varphi$ and $\psi$ are formulas, then $\neg\varphi$, $(\varphi \wedge \psi)$, $(\varphi \vee \psi)$, $(\varphi \Rightarrow \psi)$, $(\varphi \Leftrightarrow \psi)$ are also formulas.
3. If $\varphi$ is a formula and $x$ a variable, then $(\forall x\ \varphi)$ and $(\exists x\ \varphi)$ are also formulas.
4. Anything that was not created by a finite use of the above rules 1, 2 and 3 is not a formula.

□

We will use similar convention as in propositional logic and we will not write the outward parenthesis. Also note that again the negation "is stronger than other connectives", so we $\neg\alpha$ and not $(\neg\alpha)$.

**2.2.5 Syntactic Tree of a Formula.** We form the *syntactic tree* for every formula of predicate logic similarly as we did for formulas of propositional logic. The only differences are:

- Inner vertex can be labeled by one of the two quantifiers ($\forall$ and $\exists$) followed by a variable. Such vertex has one son (i.e. a quantifier is considered as "unary").
- We form also a syntactic tree of any atomic formula: The root of this subtree is labeled by the corresponding predicate symbol and it has as many sons as is its arity. The subtrees of the sons correspond to the syntactic trees of the terms written from the left to the right.

**2.2.6 Subformulas** A *subformula* of a formula $\varphi$ is any substring of $\varphi$ which is itself a correctly formed formula. $\square$

Given a formula $\beta = (\forall x\ (P(x) \vee Q(x,y))) \Rightarrow R(a,x)$. Then $\beta$ has subformulas $P(x)$, $Q(x,y)$, $R(a,x)$, $P(x) \vee Q(x,y)$, $\forall x\,(P(x) \vee Q(x,y))$, and finally itself $(\forall x\,(P(x) \vee Q(x,y))) \Rightarrow R(a,x)$.

In other words: A subformula of a formula $\varphi$ is any string that corresponds to a subtree of the syntactic tree of the formula $\varphi$, determined by some vertex labeled by a predicate symbol, by a logical connective, or by a quantifier.

**2.2.7 Bound and Free Variables.** According to the definition of a formula in 2.2.4 the following string $(\forall x\ P(x)) \Rightarrow (\exists y\ R(x,y))$ is a correctly formed formula of predicate logic. This formula was constructed using $\Rightarrow$ from two formulas, $\forall x\ P(x)$ and $\exists y\ R(x,y)$. One can see that the variable $x$ in $\forall x\ P(x)$ has nothing in common with the variable $x$ in $\exists y\ R(x,y)$. To understand better the difference between two different occurrences of $x$ as in the above example we introduce:

**Definition.** We say that an occurrence of a variable $x$ is *bounded* in a formula $\varphi$, provided there is a vertex labeled by a quantifier with this variable $x$ on the path from the leaf labeled by this $x$ to the root (backwards) in the syntactic tree. Otherwise we speak of a *free* occurrence of the variable $x$. $\square$

**2.2.8 Sentence.** A formula that does not have free variables is called a *closed formula* or a *sentence*. A formula that does not have bounded variables is called an *open formula*.

Note that there are formulas that are both closed and open; indeed, any formula that does not have variables (only constant symbols) is at the same time closed and open. Obviously, the formula above contains a variable, $x$, which has both free and bounded occurrence.

## 2.3 Semantics of Predicate Logic

Similarly as in propositional logic we need to study what does it mean that a predicate logic formula is true or false. In propositional logic it suffices to declare which of the logical variables are true (and which are false) and we easily get the truth value of the whole formula. The truthfulness was then established by a truth valuation, i.e. a mapping from the set of all logical variables into $\{0,1\}$. In predicate logic the situation is considerably more difficult; the crucial notion will be an *interpretation* which will help us to find out which sentence (a closed formula) is true and which is false in the given interpretation.

This section follows the structure of corresponding section about propositional logic.

**2.3.1 Informal Description of an Interpretation** Let us start with an example. Given a formula

$$\forall x\,(S(x) \Rightarrow S(f(f(x))))$$

where $S$ is a unary predicate symbol and $f$ is a unary functional symbol (obviously, $x$ is a variable).

The above formula can have several meanings. We state two of them.

1. Objects are people. $S$ represents the property “to be alive”. The function $f$ assigns to every person his/her father. The formula corresponds to the sentence: “Everybody who is alive has a grandfather from the father side who is also alive.”. And, of course, this is a false statement.

2. Objects are natural numbers. $S$ represents the property “to be an even number”, $f$ assigns to every natural number its successor. The formula corresponds to the sentence: “For every even natural number the successor of its successor is also an even number.” And, of course, it is a true statement.

It is evident from the example above that in order to decide whether a given formula is true or false we need to know the "meaning" of all special symbols. That is the meaning of all predicate symbols, constant symbols, and functional symbols, and what our "word" is, i.e. from what set our objects will be taken. Moreover, we must have only a sentence (e.g. for the following formula $x > 5$ we cannot decide whether it is true or false unless $x$ is bounded by one of the quantifiers).

**2.3.2 Formal Definition of an Interpretation.** An *interpretation* of predicate logic with the set of predicate symbols $\mathsf{Pred}$, the set of constant symbols $\mathsf{Cons}$, and the set of functional symbols $\mathsf{Func}$ is a pair $\langle U, [\![-]\!]\rangle$, where

- $U$ is a non-empty set called universe of domain;
- $[\![-]\!]$ is an assignment which
  1. to every predicate symbol $P \in \mathsf{Pred}$ of arity $n$ it assigns a subset $[\![P]\!]$ of $U^n$, ( we will see later that it is in fact an $n$-ary relation on the set $U$),
  2. to every constant symbol $a \in \mathsf{Cons}$ it assigns an element of $U$; we denote it by $[\![a]\!]$,
  3. to every functional symbol $f \in \mathsf{Func}$ of arity $n$ it assigns a mapping from the set $U^n$ into $U$; we denote it by $[\![f]\!]$.

**2.3.3 Informal Definition of Truth Value of a Sentence in an Interpretation.** First, we have to interpret a term in a given interpretation. If we know what should be substituted for a variable, we can then find the value of the term similarly as evaluation of algebraic expressions (values of constants are given by the interpretation as well as the meaning of the functional symbols).

An atomic formula $P(t_1, \dots, t_n)$ is true if the $n$-touple $(o_1, \dots, o_n)$ has the property $[\![P]\!]$, where $o_i$, $i = 1, \dots, n$ are values of terms $t_1, \dots, t_n$.

If we know the truth value of formulas $\alpha$ and $\beta$, then the truth value of $\neg\alpha$, $\alpha \wedge \beta$, $\alpha \vee \beta$, $\alpha \Rightarrow \beta$ and $\alpha \Leftrightarrow \beta$ are defined as in the propositional logic.

If $x$ is a variable and $\alpha$ is a formula with free variable $x$, then

- $\forall x\, \alpha$ is true if and only if for every $d \in U$ substituting $d$ for $x$ yields a true formula.
- $\exists x\, \alpha$ is true if and only if there exists at least one $d \in U$ such that if we substitute $d$ for $x$ we get a true formula.

**2.3.4 A Model of a Sentence** Given a sentence $\varphi$. Any interpretation in which $\varphi$ is true is called a *model* of $\varphi$. □

So, the second interpretations from 2.3.1 is a model of the sentence $\alpha = \forall x\, (S(x) \Rightarrow S(n(n(x)))$, whereas the first one is a model of $\neg\alpha$.

**2.3.5 A Tautology, a Contradiction, a Satisfiable Sentence.** Similarly as in propositional logic we can define:

**Definition.** A sentence is called a *tautology* provided it is true in every interpretation; it is called a *contradiction* provided it is false in every interpretation. A sentence is *satisfiable* provided there is at least one interpretation in which the formula is true. □

We can reformulate the above definition as follows: A satisfiable sentence is a sentence that has a model; a contradiction is a sentence that does not have a model; and a tautology is a sentence for which every interpretation is a model.

Note that there is infinitely many different interpretations of an even simple sentence as $\forall x\, P(x)$. Indeed, on any set we can define a subset corresponding to $P$. Hence, in predicate logic there is nothing like "truth table" for a sentence.

**Examples.** Given a unary predicate $P$ and a constant $a$. Then

1. $\alpha = (\forall x\, P(x)) \vee \neg(\forall x\, P(x))$, $\beta = (\forall x\, P(x)) \Rightarrow (\exists x\, P(x))$, $\gamma = \neg(\forall x\, P(x)) \Leftrightarrow (\exists x\, \neg P(x))$, and $\delta = \neg(\exists x\, P(x)) \Leftrightarrow (\forall x\, \neg P(x))$ are tautologies.
2. $\mu = (\exists x\, P(x) \wedge (\forall x\, \neg P(x))$ is a contradiction. Also the negation of any tautology is a contradiction.
3. $\alpha = P(a)$, $\beta = (\exists x\, P(x)) \Rightarrow P(a)$ are satisfiable sentences which are not tautologies.

Note that $x$ is a variable, otherwise the above examples were not syntactically correct formulas.

### 2.3.6 Tautological Equivalence.

**Definition.** We say that two sentences $\varphi$ and $\psi$ are *tautologically equivalent* (also *semantically equivalent*), if they are either both true or both false in every interpretation. □

**2.3.7 Examples.** Let us give a couple of examples of tautologically equivalent sentences that are typical for predicate logic.

1. $\forall x\, \forall y\, Q(x,y) \models\!\!\!\!\dashv \forall y\, \forall x\, Q(x,y)$,
2. $\exists x\, \exists y\, Q(x,y) \models\!\!\!\!\dashv \exists y\, \exists x\, Q(x,y)$.
3. $\neg(\forall x\, P(x)) \models\!\!\!\!\dashv (\exists x\, \neg P(x))$;
4. $\neg(\exists x\, P(x)) \models\!\!\!\!\dashv (\forall x\, \neg P(x))$;
5. $(\forall x\, P(x)) \vee (\forall y\, Q(y)) \models\!\!\!\!\dashv \forall x\, \forall y\, (P(x) \vee Q(y))$;
6. $(\exists x\, P(x)) \wedge (\exists y\, Q(y)) \models\!\!\!\!\dashv \exists x\, \exists y\, (P(x) \wedge Q(y))$.

**2.3.8 Semantical consequence.** Similarly as in the propositional logic we will introduce the notion of semantical consequence to capture correct entailments in predicate logic.

First let us say that a set of sentences $S$ is *true* in an interpretation $\langle U, [\![-]\!]\rangle$, if every sentence from $S$ is true in $\langle U, [\![-]\!]\rangle$.

**Definition.** We say that a sentence $\varphi$ is a *semantical consequence* of a set of sentences $S$, (or also an *entailment* of the set $S$, or that $\varphi$ *semantically follows* from the set $S$), provided $\varphi$ is true in every interpretation $\langle U, [\![-]\!]\rangle$ in which the set $S$ is true. We denote this fact by

$$S \models \varphi.$$

□

In other words, a sentence $\varphi$ *is not* a semantical consequence of a set of sentences $S$, provided that there is a model of the set $S$ which is not a model of $\varphi$. That is, there exists an interpretation $\langle U, [\![-]\!]\rangle$, in which every sentence of $S$ is true and the sentence $\varphi$ is false. Hence, the notion is similar to the semantical consequence in propositional logic, the only difference is that in propositional logic we deal with truth valuations, and in predicate logic we have interpretations.

**2.3.9 Example.** Let us decide whether $N \models \varphi$ holds, where $N = \{\forall x\, (P(x) \Rightarrow Q(x)), P(a)\}$ and $\varphi = \exists x\, Q(x)$.

**Solution.** Take any interpretation $\langle U, [\![-]\!]\rangle$, in which both the sentences $\forall x\, (P(x) \Rightarrow Q(x))$ and $P(a)$ are true. Since the formula $P(a)$ is true, the element $c = [\![a]\!]$ has the property $[\![P]\!]$. It means that $c \in [\![P]\!]$. Since also the formula $\forall x\, (P(x) \Rightarrow Q(x))$ is true in $\langle U, [\![-]\!]\rangle$ and the element $c$ has the property $[\![P]\!]$, the element $c$ has also the property $[\![Q]\!]$. It means that $c \in [\![Q]\!]$. Therefore $[\![Q]\!] \neq \emptyset$, thus the formula $\exists x\, Q(x)$ is true in $\langle U, [\![-]\!]\rangle$.

We have shown that $\varphi$ is a semantical consequence of the set $N$.