# 7. Predicate Logic

## Predicate Formulas: Syntax

"Predicate formulas" generalize the concept of a propositional formula defined in Handout 1.

A *predicate signature* is a set of symbols of two kinds—*object constants* and *predicate constants*—with a nonnegative integer, called the *arity*, assigned to every predicate constant. A predicate constant is said to be *propositional* if its arity is 0. Propositional constants are similar to atoms in propositional logic. A predicate constant is *unary* if its arity is 1, and *binary* if its arity is 2. For instance, we can define a predicate signature

$$\{a, P, Q\} \tag{1}$$

by saying that $a$ is an object constant, $P$ is a unary predicate constant, and $Q$ is a binary predicate constant.

Take a predicate signature $\sigma$ that does not include any of the following symbols:

- the *(object) variables* $x, y, z, x_1, y_1, z_1, x_2, y_2, z_2, \ldots$,
- the propositional connectives,
- the *universal quantifier* $\forall$ and the *existential quantifier* $\exists$,
- the parentheses and the comma.

The alphabet of predicate logic consists of the elements of $\sigma$ and of the four groups of additional symbols listed above. A *string* is a finite string of symbols in this alphabet.

A *term* is an object constant or an object variable. A string is called an *atomic formula* if it is a propositional constant or has the form

$$R(t_1, \ldots, t_n)$$

where $R$ is a predicate constant of arity $n$ ($n > 0$) and $t_1, \ldots, t_n$ are terms. For instance, if the underlying signature is (1) then $P(a)$ and $Q(a, x)$ are atomic formulas.

We define when a string is a *(predicate) formula* recursively, as follows:

- every atomic formula is a formula,
- both 0-place connectives are formulas,
- if $F$ is a formula then $\neg F$ is a formula,
- for any binary connective $\odot$, if $F$ and $G$ are formulas then $(F \odot G)$ is a formula,
- for any quantifier $K$ and any variable $v$, if $F$ is a formula then $KvF$ is a formula.

For instance, if the underlying signature is (1) then

$$(\neg P(a) \vee \exists x(P(x) \wedge Q(x, y)))$$

is a formula.

When we write predicate formulas, we will drop some parentheses and use other abbreviations introduced in Handout 1. A string of the form $\forall v_1 \cdots \forall v_n$ $(n \geq 0)$ will be written as $\forall v_1 \cdots v_n$, and similarly for the existential quantifier.

An occurrence of a variable $v$ in a formula $F$ is *bound* if it occurs in a part of $F$ of the form $KvG$; otherwise it is *free* in $F$. For instance, in the formula

$$\exists y P(x, y) \wedge \neg \exists x P(x, x) \tag{2}$$

the first occurrence of $x$ is free, and the other three are bound; both occurrences of $y$ in (2) are bound. We say that $v$ is *free (bound)* in $F$ if some occurrence of $v$ is free (bound) in $F$. A formula without free variables is called a *closed formula*, or a *sentence*.

## Representing English Sentences by Predicate Formulas

Before we continue the study of the syntax of predicate logic, it is useful to get some experience in translating sentences from English into the language of predicate formulas. The translation exercises below are different from the other problems in that they are not precisely stated mathematical questions: there is no way to prove, in the mathematical sense, that a translation is adequate (at least until the semantics of predicate logic is defined). There is sometimes no clear-cut difference between good and bad translations; it may happen that one translation is somewhat less adequate than another but still satisfactory.

In these translation exercises, the underlying signature is (1). We will think of object variables as ranging over the set $\mathbf{N}$ of nonnegative integers, and interpret the signature as follows:

- $a$ represents the number 10,
- $P(x)$ represents the condition "$x$ is a prime number,"
- $Q(x, y)$ represents the condition "$x$ is less than $y$."

As an example, the sentence *All prime numbers are greater than* $x$ can be represented by the formula

$$\forall y(P(y) \to Q(x, y)). \tag{3}$$

In the following two problems, represent the given English sentences by predicate formulas.

**7.1** (a) There is a prime number that is less than 10. (b) $x$ equals 0. (c) $x$ equals 9.

**7.2** There are infinitely many prime numbers.

## Substitution

Let $F$ be a formula and $v$ a variable. The result of the *substitution* of a term $t$ for $v$ in $F$ is the formula obtained from $F$ by replacing each free occurrence of $v$ by $t$. When we intend to consider substitutions for $v$ in a formula, it is convenient to denote this formula by an expression like $F(v)$; then we can denote the result of the substitution of a term $t$ for $v$ in this formula by $F(t)$. For instance, if we denote formula (2) by $F(x)$ then $F(a)$ stands for

$$\exists y P(a, y) \wedge \neg \exists x P(x, x).$$

Let $F(x)$ be formula (3) proposed above as a translation for the condition "all prime numbers are greater than $x$." A formula of the form $F(t)$, where $t$ is a term, *usually* expresses the same condition applied to the value of $t$. For instance, $F(a)$ is

$$\forall y(P(y) \to Q(a, y)),$$

which means that all prime numbers are greater than 10; $F(z_2)$ is

$$\forall y(P(y) \to Q(z_2, y)),$$

which means that all prime numbers are greater than $z_2$. There is one exception, however. The formula $F(y)$, that is,

$$\forall y(P(y) \to Q(y, y)),$$

expresses the (incorrect) assertion "every prime number is less than itself." The problem with this substitution is that $y$, when substituted for $x$ in $F(x)$, is "captured" by a quantifier. To express the assertion "all prime numbers are greater than $y$" by a predicate formula, we would have to use a bound variable different from $y$ and write, for instance,

$$\forall z(P(z) \to Q(y, z)).$$

To distinguish "bad" substitutions, as in the last example, from "good" substitutions, we introduce the following definition. A term $t$ is *substitutable* for a variable $v$ in a formula $F$ if

- $t$ is a constant, or
- $t$ is a variable $w$, and no part of $F$ of the form $KwG$ contains an occurrence of $v$ which is free in $F$.

For instance, $a$ and $z_2$ are substitutable in (3) for $x$, and $y$ is not substitutable.

## Predicate Formulas: Semantics

The semantics of propositional formulas described in Handout 1 defined which truth value $F^I$ is assigned to a propositional formula $F$ by an interpretation $I$. Our next goal is to extend this definition to predicate formulas. First we need to extend the definition of an interpretation to predicate signatures.

An *interpretation* $I$ of a predicate signature $\sigma$ consists of

- a nonempty set $|I|$, called the *universe* of $I$,
- for every object constant $c$ of $\sigma$, an element $c^I$ of $|I|$,
- for every propositional constant $R$ of $\sigma$, an element $R^I$ of $\{\mathsf{f}, \mathsf{t}\}$,
- for every predicate constant $R$ of $\sigma$ of arity $n > 0$, a function $R^I$ from $|I|^n$ to $\{\mathsf{f}, \mathsf{t}\}$.

For instance, the second paragraph of the section on representing English sentences by predicate formulas (p. 3) can be viewed as the definition of an interpretation of signature (1). For this interpretation $I$,

$$
\begin{aligned}
&|I| = \mathbf{N}, \\
&a^I = 10, \\
&P^I(n) = \begin{cases} \mathsf{t}, & \text{if } n \text{ is prime,} \\ \mathsf{f}, & \text{otherwise,} \end{cases} \\
&Q^I(m,n) = \begin{cases} \mathsf{t}, & \text{if } m < n, \\ \mathsf{f}, & \text{otherwise.} \end{cases}
\end{aligned} \tag{4}
$$

The semantics of predicate logic introduced below defines the truth value $F^I$ only for the case when $F$ is a *sentence*. As in propositional logic, the definition is recursive. For propositional constants, there is nothing to define: the truth value $R^I$ is part of the interpretation $I$. For other atomic sentences, we can define

$$R(t_1, \ldots, t_n)^I = R^I(t_1^I, \ldots, t_n^I).$$

(Since $R(t_1, \ldots, t_n)$ is a sentence, each term $t_i$ is an object *constant*, and consequently $t_i^I$ is part of $I$). For propositional connectives, we can use the same clauses as in propositional logic. But the case of quantifiers presents a difficulty. For the existential quantifier, for instance, one possibility would be to define:

$$\exists w F(w)^I = \mathsf{t} \text{ iff, for some object constant } c,\ F(c)^I = \mathsf{t}.$$

But this formulation is unsatisfactory: it disregards the fact that some elements of the universe $|I|$ may be not represented by object constants. For instance, in the example above, 10 is the only element of the universe $\mathbf{N}$ for which there is a corresponding object constant in signature (1); we expect to find that

$$(\exists x Q(x, a))^I = \mathsf{t}$$

(there exists a number that is less than 10) although

$$Q(a, a)^I = \mathsf{f}$$

(10 does not have this property). Because of this difficulty, some additional work is needed before we define $F^I$.

Consider an interpretation $I$ of a predicate signature $\sigma$. For any element $\xi$ of its universe $|I|$, select a new symbol $\xi^*$, called the *name* of $\xi$. By $\sigma^I$ we

denote the predicate signature obtained from $\sigma$ by adding all names $\xi^*$ as additional object constants. For instance, if $\sigma$ is (1) and $I$ is (4) then

$$\sigma^I = \{a, 0^*, 1^*, 2^*, \ldots, P, Q\}.$$

The interpretation $I$ can be extended to the new signature $\sigma^I$ by defining

$$(\xi^*)^I = \xi$$

for all $\xi \in |I|$. We will denote this interpretation of $\sigma^I$ by the same symbol $I$.

We will define recursively the truth value $F^I$ that is *assigned* to $F$ by $I$ for every sentence $F$ of the extended signature $\sigma^I$; that includes, in particular, every sentence of the signature $\sigma$. For any propositional constant $R$, $R^I$ is part of the interpretation $I$. Otherwise, we define:

- $R(t_1, \ldots, t_n)^I = R^I(t_1^I, \ldots, t_n^I)$,
- $\bot^I = \mathsf{f}$, $\top^I = \mathsf{t}$,
- $(\neg F)^I = \neg(F^I)$,
- $(F \odot G)^I = \odot(F^I, G^I)$ for every binary connective $\odot$,
- $\forall w F(w)^I = \mathsf{t}$ iff, for all $\xi \in |I|$, $F(\xi^*)^I = \mathsf{t}$,
- $\exists w F(w)^I = \mathsf{t}$ iff, for some $\xi \in |I|$, $F(\xi^*)^I = \mathsf{t}$.

As in propositional logic, we say that $I$ *satisfies* $F$, and write $I \models F$, if $F^I = \mathsf{t}$.

**7.3** Show that interpretation (4) satisfies $\exists x Q(x, a)$.

**7.4** Let $F(x)$ be formula (3). Show that, for every $n \in \mathbf{N}$, (4) satisfies $F(n^*)$ iff $n < 2$.

## Satisfiability

If there exists an interpretation satisfying a sentence $F$, we say that $F$ is *satisfiable*. A set $\Gamma$ of sentences is *satisfiable* if there exists an interpretation that satisfies all sentences in $\Gamma$.

In each of the following problems, determine whether the given set of sentences is satisfiable. We assume that $P$ and $Q$ are predicate constants; their arities will be every time clear from the context.

**7.5** (a) $P(a)$, $\exists x \neg P(x)$. (b) $P(a)$, $\forall x \neg P(x)$.

**7.6** $\forall x \exists y P(x, y)$, $\forall x \neg P(x, x)$.

**7.7** $\forall x \exists y P(x, y)$, $\forall x \neg P(x, x)$, $\forall xyz((P(x, y) \wedge P(y, z)) \rightarrow P(x, z))$.

## Entailment

A set $\Gamma$ of sentences *entails* a sentence $F$, or is a *logical consequence* of $\Gamma$ (symbolically, $\Gamma \models F$), if every interpretation that satisfies all sentences in $\Gamma$ satisfies $F$.

**7.8** Determine whether the sentences

$$\exists x P(x),\ \exists x Q(x)$$

entail

$$\exists x (P(x) \wedge Q(x)).$$

A sentence $F$ is *logically valid* if every interpretation satisfies $F$. This concept is similar to the notion of a tautology.

The *universal closure* of a formula $F$ is the sentence $\forall v_1 \cdots v_n F$, where $v_1, \ldots, v_n$ are all free variables of $F$. About a formula with free variables we say that it is *logically valid* if its universal closure is logically valid. A formula $F$ is *equivalent* to a formula $G$ (symbolically, $F \Leftrightarrow G$) if the formula $F \leftrightarrow G$ is logically valid.

**7.9** For each of the formulas

$$P(x) \rightarrow \exists x P(x),$$
$$P(x) \rightarrow \forall x P(x)$$

determine whether it is logically valid.

## Prenex Normal Form

A *prenex* formula is a formula of the form

$$K_1 v_1 \cdots K_n v_n F,$$

where $K_1, \ldots, K_n$ $(n \geq 0)$ are quantifiers, $v_1, \ldots, v_n$ are object variables, and $F$ is a formula without quantifiers. According to the *prenex normal form theorem*, any formula is equivalent to a prenex formula.

To find this "prenex form" of a given formula, we first eliminate $\leftrightarrow$ from it in favor of other connectives. A prenex form of a formula that does not contain $\leftrightarrow$ can be found by applying a series of "prenex operations," described below, to parts of that formula. By $F(v)$ and $G$ we denote formulas such that $v$ is not free in $G$; $K$ is a quantifier; $\forall^-$ stands for $\exists$, and $\exists^-$ stands for $\forall$. The prenex operations are:

- replace $\neg KvF(v)$ with $K^-v\neg F(v)$;
- replace $KvF(v) \wedge G$ with $Kv(F(v) \wedge G)$;
- replace $G \wedge KvF(v)$ with $Kv(G \wedge F(v))$;
- replace $KvF(v) \vee G$ with $Kv(F(v) \vee G)$;
- replace $G \vee KvF(v)$ with $Kv(G \vee F(v))$;
- replace $KvF(v) \rightarrow G$ with $K^-v(F(v) \rightarrow G)$;
- replace $G \rightarrow KvF(v)$ with $Kv(G \rightarrow F(v))$;
- replace $KvF(v)$ with $KwF(w)$, where $w$ is a variable that does not occur in $KvF(v)$.

For instance, we can find a prenex formula equivalent to

$$\exists xP(x) \rightarrow \exists xQ(x)$$

by rewriting it as

$$\forall x(P(x) \rightarrow \exists xQ(x)),$$

then as

$$\forall x(P(x) \rightarrow \exists yQ(y)),$$

and finally as

$$\forall x\exists y(P(x) \rightarrow Q(y)).$$

**7.10** Find a prenex formula equivalent to

$$\exists xP(x) \leftrightarrow \forall xQ(x).$$

## Function Symbols and Equality: Syntax of First-Order Logic

Predicate logic as defined at the beginning of this handout is somewhat more restricted than what is usually called "first-order logic," and now our goal is to remove these restrictions. First, we will generalize the notion of a term. In addition to object constants and object variables, we will allow terms to be formed using symbols for functions, "function constants." Second, we will add the equal sign to the language, and equalities will be included as a new kind of atomic formulas.

Our most general notion of a signature is defined as follows. A *signature* is a set of symbols of two kinds—*function constants* and *predicate constants*—with a nonnegative integer, called the *arity*, assigned to each symbol. An *object constant* is a function constant of arity 0. A function constant is *unary* if its arity is 1, and *binary* if its arity is 2. *Propositional constants*, as well as *unary* and *binary* predicate constants, are defined as above. *Terms* are defined recursively, as follows:

- every object constant is a term,
- every object variable is a term,
- for every function constant $h$ of arity $n$ ($n > 0$), if $t_1, \ldots, t_n$ are terms then so is $h(t_1, \ldots, t_n)$.

In first-order logic, there are three kinds of *atomic formulas*: propositional constants, strings of the form $R(t_1, \ldots, t_n)$ where $R$ is a predicate constant of arity $n$ ($n > 0$) and $t_1, \ldots, t_n$ are terms, and strings of the form $(t_1 = t_2)$ where $t_1$, $t_2$ are terms. Given this set of atomic formulas, the definition of *(first-order) formulas* is the same as the definition of predicate formulas at the beginning of this handout.

For any terms $t_1$ and $t_2$, $t_1 \neq t_2$ stands for the formula $\neg(t_1 = t_2)$.

The definitions of *free* and *bound* occurrences of a variable in a formula remain the same as before. (Note that these concepts apply to a formula, but not to a term. Sometimes, all variables occurring in a term are considered its "free variables.") The *substitution* of a term for a variable is also defined as before. The definition of a substitutable term, in the presence of function symbols, is stated as follows: A term $t$ is *substitutable* for a variable $v$ in a formula $F$ if, for each variable $w$ occurring in $t$, no part of $F$ of the form $KwG$ contains an occurrence of $v$ which is free in $F$.

## Function Symbols and Equality: Semantics of First-Order Logic

To generalize the notion of an interpretation to arbitrary signatures, we need to say how one is allowed to interpret a function constant of arity $n > 0$. Such a symbol can be interpreted as any total function of $n$ variables whose arguments and values come from the universe of the interpretation. Thus an *interpretation* $I$ of a signature $\sigma$ consists of

- a nonempty set $|I|$, called the *universe* of $I$,
- for every object constant $c$ of $\sigma$, an element $c^I$ of $|I|$,
- for every function constant $h$ of $\sigma$ of arity $n > 0$, a function $h^I$ from $|I|^n$ to $|I|$,
- for every propositional constant $R$ of $\sigma$, an element $R^I$ of $\{\mathsf{f}, \mathsf{t}\}$,
- for every predicate constant $R$ of $\sigma$ of arity $n > 0$, a function $R^I$ from $|I|^n$ to $\{\mathsf{f}, \mathsf{t}\}$.

As an example, consider the *signature of first-order arithmetic*

$$\{a, s, f, g\}, \tag{5}$$

where $a$ is an object constant (intended to represent 0), $s$ is a unary function constant (for the successor function), and $f$, $g$ are binary function constants (for addition and multiplication). Since this signature includes no predicate constants, its only atomic formulas are equalities. The intended interpretation $I$ of (5) is defined as follows:

$$\begin{aligned} &|I| = \mathbf{N}, \\ &a^I = 0, \\ &s^I(n) = n + 1, \\ &f^I(m, n) = m + n, \\ &g^I(m, n) = m \cdot n. \end{aligned} \tag{6}$$

**7.11** Represent the following English sentences by first-order formulas:

- There exists at most one $x$ such that $P(x)$.
- There exists exactly one $x$ such that $P(x)$.
- There exist at least two $x$ such that $P(x)$.

- There exist at most two $x$ such that $P(x)$.
- There exist exactly two $x$ such that $P(x)$.

Now let us go back to the semantics of first-order logic. For the definition of $F^I$ to be applicable in the presence of function constants of arity $> 0$, we need to extend the notation $t^I$ from object constants to arbitrary variable-free terms of the signature $\sigma^I$, and to add a clause for equality. The value $t^I$ *assigned* by $I$ to $t$ is defined recursively by the formula

$$h(t_1, \ldots, t_n)^I = h^I(t_1^I, \ldots, t_n^I).$$

The additional clause in the definition of satisfaction reads as follows:

- $(t_1 = t_2)^I = \mathsf{t}$ iff $t_1^I = t_2^I$.

The definitions of satisfaction and of all other semantic concepts introduced above for formulas of a predicate signature apply to formulas with function constants and equality without any changes.

**7.12** For each of the following sentences determine whether it is satisfiable:

- $a = b$,
- $\forall xy(x = y)$,
- $\forall xy(x \neq y)$.