# Logic for Computer Science - Lecture Notes

Alexandru Ioan Cuza University, Iași
Faculty of Computer Science
University Year 2021-2022

Ștefan Ciobâcă
Andrei Arusoaie
Rodica Condurache
Cristian Masalagiu

# Contents

# Chapter 1

# Motivation and introduction

First-order logic, what we will be studying next, is an extension of propositional logic, extension that brings more expressivity. The additional expressivity is necessary in order to model certain statements that cannot be expressed in propositional logic.

In propositional logic, we cannot express naturally the following statement: *All men are mortal.*

To model a statement in propositional logic, we identify the atomic propositions. Then we associate to each atomic proposition a propositional variable. The atomic propositions are the propositions that cannot be split into one or more smaller propositions, linked among them by the logical connectives of propositional logic: $\neg$, $\wedge$, $\vee$, $\rightarrow$ and $\leftrightarrow$.

We notice that the statement *All men are mortal* cannot be decomposed into smaller statements linked among them by the logical connectives of propositional logic, as is described above. Therefore, in propositional logic, the statement is atomic. So we associate to the entire statement a propositional variable $\mathtt{p} \in A$.

Let us now model the statement *Socrates is a man.* Obviously, to this second statement we must associate another propositional variable $\mathtt{q} \in A$. Let us assume that $\mathtt{p}$ and $\mathtt{q}$ are true. Formally, we work in a truth assignment $\tau : A \rightarrow B$ where $\tau(\mathtt{p}) = 1$ and $\tau(\mathtt{q}) = 1$. Can we draw the conclusion that *Socrates is mortal* in the truth assignment $\tau$?

No, because to the statement *Socrates is mortal* we should associate a third propositional variable $\mathtt{r} \in A$. We cannot draw any conclusion on $\tau(\mathtt{r})$ from $\tau(\mathtt{p}) = 1$ and $\tau(\mathtt{q}) = 1$. So, from the semantics of propositional logic, we cannot draw the conclusion that $\mathtt{r}$ is true in any truth assignment that makes

both p and q true. This is despite the fact that, in any world where *All men are mortal* and *Socrates is a man*, we can draw the conclusion that *Socrates* is mortal without failure. This difference between reality and our modelling indicates that our modelling is not sufficient for our purposes.

First-order logic includes, in addition to propositional logic, the notion of *quantifier* and the notion of *predicate*. The universal quantifier is denoted by $\forall$ and the existential quantifier is denoted by $\exists$.

A predicate is a statement whose truth value depends on zero or more parameters. For example, for the statements above, we will be using two predicates: Man and Mortal. The predicate Man is the predicate that denotes the quality of being a man: Man(x) is true iff x is a man. The predicate Mortal is true when its argument is mortal. As the predicates above have only one argument/parameter, they are called *unary* predicates. Predicates generalize propositional variables by the fact that they can take arguments. Actually, propositional variable can be regarded as predicates with no arguments.

In this way, the statement *All men are mortal* will be modelled by the formula

$$(\forall x.(\mathtt{Man}(x) \rightarrow \mathtt{Mortal}(x))),$$

which is read as follows: *for any* x*, if* Man *of* x*, then* Mortal *of* x. The statement *Socrate is a men* shall be modelled by the formula $\mathtt{Man}(s)$, where s is a *constant* that denotes Socrates, just like 0 denotes the natural number *zero*. For example, Man(s) is true (as s stands for a particular man – Socrates), but Man(l) is false if l is a constant standing for the dog *Lassie*.

The statement *Socrates is mortal* shall be represented by Mortal(s) (recall that the constant s stands for Socrates). The statement Mortal(s) is true, as Socrates is mortal; likewise, the statement Mortal(l) is also true.

We shall see that in first-order logic, the formula Mortal(s) is a logical consequence of the formulae $(\forall x.(\mathtt{Man}(x) \rightarrow \mathtt{Mortal}(x)))$ and respectively Man(s). Therefore, first-order logic is sufficiently expressive to explain theoretically the argument by which we deduce that *Socrates is mortal* from the facts that *All men are mortal* and *Socrates is a man*.

# Chapter 2

# Structures and signatures

You have certainly met already several first-order logic formulae, without necessarily knowing that you are dealing with first-order logic. Consider the following formula:

$$\varphi = \Big(\forall \mathsf{x}.(\forall \mathsf{y}.(\mathsf{x} < \mathsf{y} \rightarrow \exists \mathsf{z}.(\mathsf{x} < \mathsf{z} \wedge \mathsf{z} < \mathsf{y})))\Big).$$

The formula makes use of a binary predicate, $<$, that is defined as follows: $<(x, y)$ is true if $x$ is strictly smaller than $y$. In order to simplify our writing, we use the infixed notation $(x < y)$ instead of the prefixed notation $(<(x, y))$ for many binary predicates (including for $<$).

Is the formula $\varphi$ above true? The formula states that between any two values of the variables $\mathsf{x}, \mathsf{y}$ there is a third value, of the variable $\mathsf{z}$. The formula is true if the domain of the variables $\mathsf{x}, \mathsf{y}, \mathsf{z}$ is $\mathbb{R}$, but it is false if the domain is $\mathbb{N}$ (between any two real numbers there exists a third, but between two consecutive naturals there is no other natural number).

Generally, first-order formulas refer to a particular *mathematical structure*.

**Definition 1** (Mathematical structure)**.** *A* mathematical structure *is a tuple* $S = (D, Pred, Fun)$ *where:*

- $D$ *is a non-empty set called the* domain *of the structure;*
- *each* $P \in Pred$ *is a predicate (of a certain arity) over the set* $D$*;*
- *each* $f \in Pred$ *is a function (of a certain arity) over the set* $D$*.*

Here are a few examples of mathematical structures:

1. $(\mathbb{N}, \{<, =\}, \{+, 0, 1\})$;

The domain of the structure is the set of naturals. The structure contains two predicates: $<$ and $=$, both of arity 2. The predicate $<$ is the *smaller than* predicate on naturals, and the predicate $=$ is the *equality* predicate over natural numbers.

The structure also contains three functions. The binary function $+ : \mathbb{N}^2 \to \mathbb{N}$ is the addition function for naturals, and the functions $0 : \mathbb{N}^0 \to \mathbb{N}$ and respectively $1 : \mathbb{N}^0 \to \mathbb{N}$ are the arity 0 functions (also called constant functions or simply constants) 0 and respectively 1.

2. $(\mathbb{R}, \{<, =\}, \{+, -, 0, 1\})$;

   This structure contains two binary predicates, $<$ and $=$, as well as four functions over $\mathbb{R}$: the binary function $+$, the unary function $-$ (unary minus) and the constants $0, 1 \in \mathbb{R}$.

3. $(\mathbb{Z}, \{<, =\}, \{+, -, 0, 1\})$;

   This structure is similar to that above, but the domain is the set of integers.

4. $(B, \emptyset, \{\cdot, +, -\})$;

   This structure is a boolean algebra, where the domain is the set truth values and the functions are those that we studied in the first half of the semester. Such structures, without any predicates, are called *algebraic structures*.

5. $(\mathbb{R}, \{<\}, \emptyset)$.

   This structure contains only a predicate of arity 2 (the *less than* relation over $\mathbb{R}$) and no function. Structures without functions are called relational structures. Relational structures with a finite domain are called relational data bases and you will study them in your second year.

Whenever we want to evaluate the truth value of a first-order formula we need a mathematical structure. Recall our previous formula:

$$\varphi = \Big(\forall x.(\forall y.(x < y \to \exists z.(x < z \wedge z < y)))\Big).$$

This formula is true in the structure $(\mathbb{R}, \{<, =\}, \{+, -, 0, 1\})$ (between any two distinct real numbers there is another real number), but it is false in the structure $(\mathbb{Z}, \{<, =\}, \{+, -, 0, 1\})$ (because it is not true that between any two distinct integers there is a third integer – for example there is no such integer between two consecutive integers).

It is possible for two different structure to have a set of predicates and a set of functions with the same names. For example, the structures above, $(\mathbb{R}, \{<, =\}, \{+, -, 0, 1\})$ and respectively $(\mathbb{Z}, \{<, =\}, \{+, -, 0, 1\})$. Even if the

predicate $<\in \mathbb{R}^2$ is different from the predicate $<\in \mathbb{Z}^2$, they both have the same name: $<$.

Generally, in Mathematics and in Computer Science, we do not make any difference between a predicate and its name or between a function and its name. However, in Logic, the difference is extremely important. In particular, if we refer to the name of a function, we shall use the phrase "functional symbol" (i.e., symbol standing for a function). When we refer to the name of a predicate, we shall use the phrase "predicate symbol" (or "relational symbol"). Why is the difference between a predicate and a predicate symbol important? Because we shall need to associate to the same predicate symbol several predicates, similarly to how we can associate several values to a program variable in an imperative language.

When we are interested only in the function and predicate names (not the function or predicates themselves), we work with signatures:

**Definition 2** (Signature)**.** *A* signature $\Sigma$ *is a tuple* $\Sigma = (\mathcal{P}, \mathcal{F})$*, where* $\mathcal{P}$ *is a set of predicate symbols and* $\mathcal{F}$ *is a set of functional symbols. Each predicate or functional symbol* $s$ *has an associate natural number called its* arity *denoted by* $ar(s)$*.*

To a signature we can associate many structures:

**Definition 3** ($\Sigma$-structure)**.** *If* $\Sigma = (\mathcal{P}, \mathcal{F})$ *is a signature, a* $\Sigma$-structure *is any structure* $S = (D, Pred, Fun)$ *so that for each predicate symbol* $P \in \mathcal{P}$*, exists a predicate* $P^S \in Pred$ *of corresponding arity, and for every functional symbol* $f \in \mathcal{F}$*, there is a function* $f^S \in Fun$ *of corresponding arity.*

**Example 4.** *Let* $\Sigma = (\{\mathtt{P}, \mathtt{Q}\}, \{\mathtt{f}, \mathtt{i}, \mathtt{a}, \mathtt{b}\})$*, where* P *and* Q *are predicate symbols of arity* $ar(\mathtt{P}) = ar(\mathtt{Q}) = 2$ *and* f, i, a, b *are function symbols having the following arrities:* $ar(\mathtt{f}) = 2$*,* $ar(\mathtt{i}) = 1$ *and* $ar(\mathtt{a}) = ar(\mathtt{b}) = 0$*.*

*We have that* $(\mathbb{R}, \{<, =\}, \{+, -, 0, 1\})$ *and* $(\mathbb{Z}, \{<, =\}, \{+, -, 0, 1\})$ *are* $\Sigma$*-structures.*

**Remark.** *As you can observe in Example 4, for predicate symbols (e.g.,* P*,* Q*) we use a different color than the color used for functional symbols (e.g.,* f*,* i*,* a*,* b*). For oredicates and and functions we use the normal font for mathematical formulas.*

To remember!

Structure = domain + predicates + functions

Signature = predicate symbols + functional symbols

To a signature $\Sigma$ we can associate many structures, which are called $\Sigma$-structures.

**Notation.** *The set of predicate symbols of arity $n$ is denoted by $\mathcal{P}_n = \{P \mid ar(P) = n\}$, and the set of functional symbols of arity $n$ is $\mathcal{F}_n = \{f \mid ar(f) = n\}$. For the particular case when $n = 0$, $\mathcal{F}_0$ is the set of constant symbols (that is, functional symbols with arity 0).*

## 2.1 Exercise sheet

**Exercise 5.** *Identify the predicates and the functions in the text below. What is their domain?*

John is a student. Any student learns Logic. Anyone learning Logic passes the exam. Any student is a person. There is a person who did not pass the exam. Therefore: not all persons are students.

**Exercise 6.** *Identify the predicates and the functions in the text below. What is their domain?*

The sum of two even numbers is even.

**Exercise 7.** *Identify the predicates and the functions in the text below. What is their domain?*

In chess, the quuen can make a move from one square to another iff the bishop or the rook can make the same move.

**Exercise 8.** *Identify the predicates and the functions in the text below. What is their domain?*

The sum of two numbers greater than zero is greater than zero.

**Exercise 9.** *Identify the predicates and the functions in the text below. What is their domain?*

The number 7 is prime.

**Exercise 10.** *Identify the predicates and the functions in the text below. What is their domain?*

Any even number greater than 2 is the sum of two primes.

**Exercise 11.** *Identify the predicates and the functions in the text below. What is their domain?*

If the Earth is flat, then $2 + 2 = 5$.

**Exercise 12.** *Identify the predicates and the functions in the text below. What is their domain?*

For any $\epsilon \in (0, \infty)$, there exists $\delta_\epsilon \in (0, \infty)$ such that for any $x \in \mathbb{R}$ with $d(x_0, x) < \delta_\epsilon$, we have $d(f(x_0), f(x)) < \epsilon$.

# Chapter 3

# The Syntax of First-Order Logic

In this chapter we present the syntax of first-order logic formula. The language of first-order logic is parameterised by a signature. A difference to propositional logic is that there are several first-order logic languages, one first-order language for each signature $\Sigma$. In propositional logic, there was just one language, $\mathbb{PL}$.

Next, we shall fix a signature $\Sigma$ that contains the predicate symbols in $\mathcal{P}$ and the functional symbols in $\mathcal{F}$.

## 3.1 The Alphabet of First-Order Logic

Just as propositional logic formulae, the formulae in first-order logic are strings of characters over a certain alphabet. Unlike propositional logic, the alphabet is now richer. The alphabet of first-order logic consists of the follows "characters":

1. the logical connectives already known: $\neg, \wedge, \vee, \rightarrow, \leftrightarrow, \bot$, as well as two new *quantifiers*: $\forall$, $\exists$;
2. variables: we will assume that a countably infinite set of variables $\mathcal{X} = \{\mathsf{x}, \mathsf{y}, \mathsf{z}, \mathsf{x}', \mathsf{y}', \mathsf{x}_1, \mathsf{z}'', \ldots\}$ is also part of the alphabet (not to be confused with propositional variables in propositional logic – they are two fundamentally different notions);
3. auxilliary symbols: $($, $)$, $.$, $,$, $($, $)$, and $,$;
4. non-logical symbols, that are specific to each signature $\Sigma = (\mathcal{P}, \mathcal{F})$: the functional symbols in $\mathcal{F}$ and the predicate symbols in $\mathcal{P}$.

## 3.2 Terms

**Definition 13.** *The set of* terms, $\mathcal{T}$, *is the smallest set having the following properties:*

1. $\mathcal{F}_0 \subseteq \mathcal{T}$ *(any constant symbol is a term);*
2. $X \subseteq \mathcal{T}$ *(any variable is a term);*
3. *if* $f \in \mathcal{F}_n$ *(with* $n > 0$*) and* $t_1, \ldots, t_n \in \mathcal{T}$*, then* $f(t_1, \ldots, t_n) \in \mathcal{T}$ *(a functional symbol of arity* $n$ *applied to* $n$ *terms is a term).*

**Remark.** *The elements of the set* $\mathcal{T}$ *are often called* $\Sigma$*-terms, because the definition of* $\mathcal{T}$ *depends on* $\Sigma$*.*

Terms are essentially built by applying functional symbols to variables and constant symbols.

**Example 14.** *Recall* $\Sigma = (\{\mathtt{P}, \mathtt{Q}\}, \{\mathtt{f}, \mathtt{i}, \mathtt{a}, \mathtt{b}\})$ *from Example 4, where* $ar(\mathtt{P}) = ar(\mathtt{Q}) = 2$*,* $ar(\mathtt{f}) = 2$*,* $ar(\mathtt{i}) = 1$*,* $ar(\mathtt{a}) = ar(\mathtt{b}) = 0$*. Here are several example of terms:* $\mathtt{a}$*,* $\mathtt{b}$*,* $\mathtt{x}$*,* $\mathtt{y}$*,* $\mathtt{x}_1$*,* $\mathtt{y}'$*,* $\mathtt{i(a)}$*,* $\mathtt{i(x)}$*,* $\mathtt{i(i(a))}$*,* $\mathtt{i(i(x))}$*,* $\mathtt{f(a,b)}$*,* $\mathtt{i(f(a,b))}$*,* $\mathtt{f(f(x,a),f(y,y))}$*.*

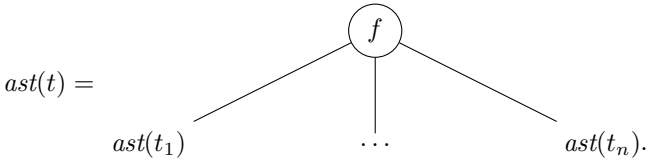
**Exercise 15.** *Identify the* $\Sigma$*-terms in the following list :*

1. $\mathtt{i(i(x))}$*;*
2. $\mathtt{i}$*;*
3. $\mathtt{f(x,x)}$*;*
4. $\mathtt{P(a,b)}$*;*
5. $\mathtt{i(a,a)}$*;*
6. $\mathtt{f(i(x),i(x))}$*;*
7. $\mathtt{f(i(x,x))}$*;*
8. $\mathtt{a(i(x))}$*.*

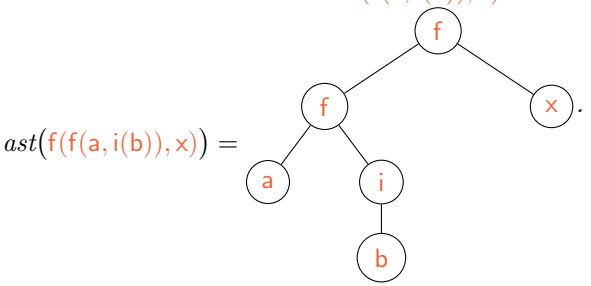
We denote terms by the letters $t, s, t_1, t_2, s_1, t'$, etc. Even if terms are usually written as a string of characters, they have an associated abstract syntax tree defined as follows:

1. if $t = c$ and $c \in \mathcal{F}_0$, then $ast(t) = \textcircled{c}$;
2. if $t = x$ and $x \in \mathcal{X}$, then $ast(t) = \textcircled{x}$;

3. if $t = f(t_1, \ldots, t_n)$ and $f \in \mathcal{F}_n$ $(n > 0)$, $t_1, \ldots, t_n \in \mathcal{T}$, then




**Remark.** *Even if formally terms are defined as strings of characters over the alphabet described above, these must be understood as being trees. In any software program that handles terms, these are stored as a rooted tree. Here is the tree associated to the term* $\mathsf{f(f(a,i(b)),x)}$*:*




**Exercise 16.** *Compute the abstract syntax trees for all terms in Example 14.*

## 3.3 Atomic formulae

**Definition 17** (Atomic formula)**.** *An* atomic formula *is any string of characters of the form* $P(t_1, \ldots, t_n)$*, unde* $P \in \mathcal{P}_n$ *, where* $P \in \mathcal{P}_n$ *is a predicate symbol of arity* $n$*, and* $t_1, \ldots, t_n \in \mathcal{T}$ *are terms.*

**Example 18.** *Continuing the previous example, we work over the signature*

$$\Sigma = (\{\mathsf{P}, \mathsf{Q}\}, \{\mathsf{f}, \mathsf{i}, \mathsf{a}, \mathsf{b}\}),$$

*where* $ar(\mathsf{P}) = ar(\mathsf{Q}) = 2$*,* $ar(\mathsf{f}) = 2$*,* $ar(\mathsf{i}) = 1$*,* $ar(\mathsf{a}) = ar(\mathsf{b}) = 0$*.*

*Here are a few examples of atomic formulae:* $\mathsf{P(a,b)}$*,* $\mathsf{P(x,y)}$*,* $\mathsf{Q\Big(i(i(x)),f(x,x)\Big)}$*,* $\mathsf{Q(a,b)}$*,* $\mathsf{P\Big(f\big(f(a,i(x)),b\big),i(x)\Big)}$*.*

**Exercise 19.** *Explain why* $\mathsf{P(a)}$*,* $\mathsf{P}$*,* $\mathsf{i(i(x))}$ *are not atomic formulae over the signature in Example 18.*

## 3.4 First-Order Formulae

**Definition 20** (First-Order Formula)**.** *The set of first-order formulae, written* $\mathbb{FOL}$*, is the smallest set with the following properites:*

1. *(base case) any atomic formula is a formula (that is* $P(t_1, \ldots, t_n) \in \mathbb{FOL}$ *for any predicate symbol* $P \in \mathcal{P}_n$ *and any terms* $t_1, \ldots, t_n$*; if* $n = 0$*, we write* $P$ *instead of* $P()$*);*
2. *(inductive cases) for any formulae* $\varphi, \varphi_1, \varphi_2 \in \mathbb{FOL}$*, for any variable* $x \in \mathcal{X}$*, we have:*
   (a) $\neg\varphi_1 \in \mathbb{FOL}$*;*
   (b) $(\varphi_1 \wedge \varphi_2) \in \mathbb{FOL}$*;*
   (c) $(\varphi_1 \vee \varphi_2) \in \mathbb{FOL}$*;*
   (d) $(\varphi_1 \to \varphi_2) \in \mathbb{FOL}$*;*
   (e) $(\varphi_1 \leftrightarrow \varphi_2) \in \mathbb{FOL}$*;*
   (f) $(\forall x.\varphi) \in \mathbb{FOL}$*;*
   (g) $(\exists x.\varphi) \in \mathbb{FOL}$*.*

**Remark.** *In Definition 20, we find the logical connectives* $\neg, \wedge, \vee, \to$ *and* $\leftrightarrow$ *from propositional logic. The predicate symbols of arity* 0 *play the role of propositional variables (for now, at the syntactic level). The constructions* $(\forall x.\varphi)$ *and* $\exists x.\varphi)$ *are new.*

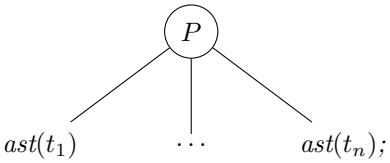
**Example 21.** *Recall* $\Sigma = (\{\mathtt{P}, \mathtt{Q}\}, \{\mathtt{f}, \mathtt{i}, \mathtt{a}, \mathtt{b}\})$ *from Example 14, where* $ar(\mathtt{P}) = ar(\mathtt{Q}) = 2$*,* $ar(\mathtt{f}) = 2$*,* $ar(\mathtt{i}) = 1$ *și* $ar(\mathtt{a}) = ar(\mathtt{b}) = 0$*.*

*Here are several example of first-order formulae:*

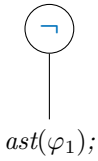
1. $\mathtt{P}(\mathtt{a},\mathtt{b})$*;*
2. $\mathtt{Q}(\mathtt{a},\mathtt{b})$*;*
3. $\mathtt{P}(\mathtt{a},\mathtt{x})$*;*
4. $\neg\mathtt{P}(\mathtt{a},\mathtt{b})$*;*
5. $(\mathtt{P}(\mathtt{a},\mathtt{b}) \wedge \neg\mathtt{Q}(\mathtt{a},\mathtt{b}))$*;*
6. $(\mathtt{P}(\mathtt{a},\mathtt{b}) \vee \neg\mathtt{Q}(\mathtt{x},\mathtt{y}))$*;*
7. $(\mathtt{P}(\mathtt{a},\mathtt{b}) \to \mathtt{P}(\mathtt{a},\mathtt{b}))$*;*
8. $((\mathtt{P}(\mathtt{a},\mathtt{b}) \to \mathtt{P}(\mathtt{a},\mathtt{b})) \leftrightarrow (\mathtt{P}(\mathtt{a},\mathtt{b}) \to \mathtt{P}(\mathtt{a},\mathtt{b})))$*;*
9. $(\forall\mathtt{x}.\mathtt{P}(\mathtt{a},\mathtt{x}))$*;*
10. $(\exists\mathtt{x}.\neg\mathtt{Q}(\mathtt{x},\mathtt{y}))$*.*

**Definition 22** (The Abstract Syntax Tree of formulae in $\mathbb{FOL}$)**.** *Formulae have an associated abstract syntax tree defined as follows:*

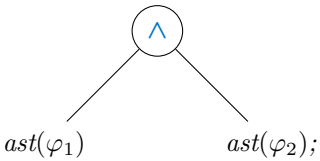
1. *if* $\varphi = P(t_1, \ldots, t_n)$, *then* $ast(\varphi) =$



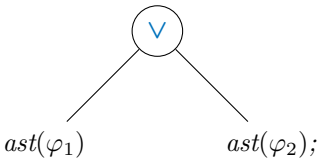

2. *if* $\varphi = \neg\varphi_1$, *then* $ast(\varphi) =$



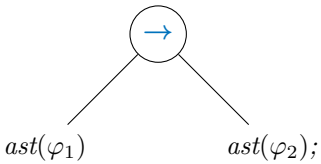

3. *if* $\varphi = (\varphi_1 \wedge \varphi_2)$, *then* $ast(\varphi) =$



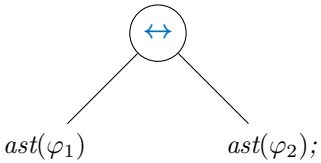

4. *if* $\varphi = (\varphi_1 \vee \varphi_2)$, *then* $ast(\varphi) =$



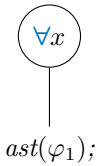

5. *if* $\varphi = (\varphi_1 \rightarrow \varphi_2)$, *then* $ast(\varphi) =$

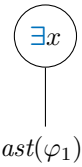



6. *if* $\varphi = (\varphi_1 \leftrightarrow \varphi_2)$, *then* $ast(\varphi) =$




7. *if* $\varphi = (\forall x.\varphi_1)$, *then* $ast(\varphi) =$



8. *if* $\varphi = (\exists x.\varphi_1)$, *then* $ast(\varphi) =$ 


**Exercise 23.** *Compute the abstract syntax tree of the formulae shown in Example 21.*

## 3.5 Model English sentences as formulae in $\mathbb{FOL}$

Next, we will explain the signature used to model in first-order logic the statments *All men are mortal*, *Socrates is a man* and respectively *Socrates is mortal*.

First, we identify the predicates in the text. We have two unary predicates *is a man* and respectively *is mortal*. We choose the predicate symbol $\mathtt{H}$ for the first predicate and the predicate symbol $\mathtt{M}$ for the second predicate. We also have one constant in the text: Socrates. We choose the functional symbol $\mathtt{s}$ (of arity 0) for this constant. Therefore, to model the statements above, we shall work in the signature

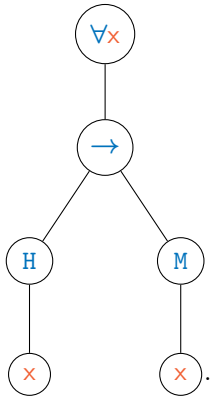
$$\Sigma = (\{\mathtt{H}, \mathtt{M}\}, \{\mathtt{s}\}),$$

where $\mathtt{H}$ and $\mathtt{M}$ are predicate symbols of arity $ar(\mathtt{H}) = ar(\mathtt{M}) = 1$, and $\mathtt{s}$ is a functional symbol of arity $ar(\mathtt{s}) = 0$.

The statement *All men are mortal* will be modelled by the first-order formula

$$(\forall \mathtt{x}.(\mathtt{H}(\mathtt{x}) \to \mathtt{M}(\mathtt{x}))),$$

whose abstract syntax tree is:

$ast\Big((\forall \mathtt{x}.(\mathtt{H}(\mathtt{x}) \to \mathtt{M}(\mathtt{x})))\Big) =$ 

The statement *Scorates is a man* shall be modelled by the atomic formula $\mathtt{H}(\mathtt{s})$, and the statement *Socrates is mortal* by the atomic formula $\mathtt{M}(\mathtt{s})$.

For the signature $\Sigma = (\{\mathtt{H}, \mathtt{M}\}, \{\mathtt{s}\})$ fixed above, there exist several possible $\Sigma$-structures. An example of a $\Sigma$-structure would be $S = (D, \{\mathtt{H}^S, \mathtt{M}^S\}, \{s^S\})$ defined as follows:

1. $D$ is the set of all beings on Earth;
2. $\mathtt{H}^S(x)$ is true for any being $x$ that is a man;
3. $\mathtt{M}^S(x)$ is true of any being $x$ (all of the elements in the domain are mortal);
4. $\mathtt{s}^S$ is Socrates (Socrates, being a being, belongs to the set $D$).

Anticipanting a little bit (we shall discuss the semantics of first-order formulae in the next lecture), all tree formulae discussed in this section, $(\forall \mathtt{x}.(\mathtt{H}(\mathtt{x}) \rightarrow \mathtt{M}(\mathtt{x})))$, $\mathtt{H}(\mathtt{s})$ and respectively $\mathtt{M}(\mathtt{s})$, are true in the structure $S$ defined above.

In fact, the quality of the argument *All men are mortal; Socrates is a man; so: Socrates is mortal* is given by the fact that the formula $\mathtt{M}(\mathtt{s})$ is necessarily true in *any* structure in which the formulae $(\forall \mathtt{x}.(\mathtt{H}(\mathtt{x}) \rightarrow \mathtt{M}(\mathtt{x})))$ and respectively $\mathtt{H}(\mathtt{s})$ are true, not just in the structure $S$ above.

## 3.6 Model arithmetic sentences as formulae in $\mathbb{FOL}$

Consider the signature $\Sigma = (\{<, =\}, \{+, -, 0, 1\})$, where $<$ and $=$ are predicate symbols of arity 2, $+$ is a functional symbol of arity 2, $-$ is a functional symbol of arity 1, and $0$ and $1$ are constant symbols.

Here are a few first-order formulae in the first-order language associated to the signature $\Sigma$:

1. $\Big(\forall \mathtt{x}.\big(\forall \mathtt{y}.\big(<(\mathtt{x}, \mathtt{y}) \rightarrow \exists \mathtt{z}.(<(\mathtt{x}, \mathtt{z}) \wedge <(\mathtt{z}, \mathtt{y}))\big)\big)\Big)$;
2. $\Big(\forall \mathtt{x}.\big(\forall \mathtt{y}.\big(\exists \mathtt{z}.(=(+(\mathtt{x}, \mathtt{y}), \mathtt{z}))\big)\big)\Big)$;
3. $\Big(\forall \mathtt{x}.(<(0, \mathtt{x}) \vee =(0, \mathtt{x}))\Big)$;
4. $\Big(\forall \mathtt{x}.\big(\exists \mathtt{y}.(=(\mathtt{x}, -(\mathtt{y})))\big)\Big)$;
5. $=(+(\mathtt{x}, \mathtt{y}), \mathtt{z})$.

Many times, in the case of binary predicate symbols and binary functional symbols, we use the infixed notation (e.g., $x<y$ instead of $<(x,y)$). In this case, we could write the formulae above as follows:

1. $\Big(\forall x.\big(\forall y.\big(x < y \rightarrow \exists z.\big(x < z \wedge z < y\big)\big)\big)\Big)$;
2. $\Big(\forall x.\big(\forall y.\big(\exists z.\big(x + y = z\big)\big)\big)\Big)$;
3. $\Big(\forall x.\big(0 < x \vee 0 = x\big)\Big)$;
4. $\Big(\forall x.\big(\exists y.\big(x = -(y)\big)\big)\Big)$;
5. $x + y = z$.

Two of the possible $\Sigma$-structures are $S_1 = (\mathbb{R}, \{<, =\}, \{+, -, 0, 1\})$ and $S_2 = (\mathbb{Z}, \{<, =\}, \{+, -, 0, 1\})$, where the predicates and functions are those known from mathematics (with the remark that $-$ is the unary minus function).

Anticipanding the next lecture, on the semantics of first-order formulae, the first formula is false in $S_2$ and true in $S_1$. The second and the fourth formula are true both in $S_1$ and in $S_2$. The third formulae is false both in $S_1$ and in $S_2$. The truth value of the fifth formula depends not only of the structure where we evalute the truth value of the formula, but also on the values of the variables $x, y, z$. Because the variables $x, y, z$ are not protected by a quantifier in formula number 5, they are called *free variables*. Formula number 5 is *satisfiable* both in the structure $S_1$ as well as in the structure $S_2$, because in both cases there are values for the variables $x, y, z$ that make the formula true (e.g. the values $1, 2, 3$ for $x, y$, and respectively $z$).

## 3.7 Exercises

**Exercise 24.** *Identify a signature for the following statements and model the statements as formulae in first-order logic over that signature.*

John is a student. Any student learns Logic. Anyone learning Logic passes the exam. Any student is a person. There is a person who did not pass the exam. Therefore: not all persons are students.

**Exercise 25.** *Consider $S = (\mathbb{R}, \{Nat, Int, Prime, Even, >\}, \{+, 0, 1, 2\})$, a structure where Nat, Int, Prime, Even are unary predicates with the following meaning: $Nat(u)$ = "u is a natural number, $Int(u)$ = "u is an integer number", $Prime(u)$ = "u is a prime" and $Even(u)$ = "u is an even number". The binary predicate $>$ is the "greater than" relation over real numbers. The*

*function $+$ is the addition function for real numbers. The constants $0, 1, 2$ are what you would expect.*

1. *Find a signature $\Sigma$ so that the structure $S$ is a $\Sigma$ structure;*
2. *Model the following statements as first-order formulae in the signature associated to the structure $S$ above:*
   (a) *Any natural number is also an integer.*
   (b) *The sum of any two natural numbers is a natural number.*
   (c) *No matter how we would choose a natural number, there is prime number that is greater than the number we chose.*
   (d) *If any natural number is a prime number, then zero is a prime number.*
   (e) *No matter how we choose a prime number, there is a prime number greater than it.*
   (f) *The sum of two even numbers is an even number.*
   (g) *Any prime number greater than 2 is odd.*
   (h) *Any prime number can be written as the sum of four prime numbers.*
   (i) *The sum of two even numbers is an odd number.*
   (j) *Any even number is the sum of two primes.*

**Exercise 26.** *Give examples of 5 terms over the signature in Exercise 25 (1.) and compute their abstract syntax tree.*

**Exercise 27.** *Give examples of 5 formulae over the signature in ExerciseExercise 25 (1.) and compute their abstract syntax tree.*

**Exercise 28.** *Compute the abstract syntax tree of the following formulae:*

1. $(\mathtt{P}(\mathtt{x}) \vee (\mathtt{P}(\mathtt{y}) \wedge \neg \mathtt{P}(\mathtt{z})))$;
2. $((\neg\neg \mathtt{P}(\mathtt{x}) \vee \mathtt{P}(\mathtt{y})) \rightarrow (\mathtt{P}(\mathtt{x}) \wedge \neg \mathtt{P}(\mathtt{z})))$;
3. $(\forall \mathtt{x}.(\forall \mathtt{y}.((\neg\neg \mathtt{P}(\mathtt{x}) \vee \mathtt{P}(\mathtt{y})) \rightarrow (\mathtt{P}(\mathtt{x}) \wedge \neg \mathtt{P}(\mathtt{z})))))$;
4. $(\forall \mathtt{x}.(\forall \mathtt{y}.((\neg\neg \mathtt{P}(\mathtt{x}) \vee \mathtt{P}(\mathtt{y})) \rightarrow (\exists \mathtt{x}.(\mathtt{P}(\mathtt{x}) \wedge \neg \mathtt{P}(\mathtt{x}))))))$;
5. $(\forall \mathtt{x}'.\neg(\forall \mathtt{x}.(\mathtt{P}(\mathtt{x}) \wedge (\exists \mathtt{y}.((\mathtt{Q}(\mathtt{x}, \mathtt{y}) \vee \neg \mathtt{Q}(\mathtt{z}, \mathtt{z})) \rightarrow (\exists \mathtt{z}'.\mathtt{P}(\mathtt{z}')))))))$.

# Chapter 4

# The Semantics of First-Order Formulae

The syntax of first-order logic tells us what is, from a syntactical point of view, a formula in first-order logic. On the other hand, the *semantics* of the logic refers to the *meaning* of formulae. The semantics of a formula (or its meaning) is a truth value. In general, just as for propositional logic, the truth value of a formula depends not only on the formula itself, but also on some extra information. In the case of first-order logic, the extra information is *mathematical structure* in which the formula is evaluated.

We recall that a signature $\Sigma = (\mathcal{P}, \mathcal{F})$ is a pair consisting of predicate symbols $\mathcal{P}$ and functional symbols $\mathcal{F}$, each symbol having an arity.

In this chapter, we work over the signature $\Sigma = (\{\mathtt{P}\}, \{\mathtt{f}, \mathtt{i}, \mathtt{e}\})$, where $\mathtt{P}$ is a predicate symbol of arity 2, and $\mathtt{f}$, $\mathtt{i}$ and $\mathtt{e}$ are function symbols of arity 2, 1 and respectively 0. Therefore, $\mathcal{P}_2 = \{\mathtt{P}\}, \mathcal{P}_1 = \emptyset, \mathcal{P}_0 = \emptyset, \mathcal{F}_2 = \{\mathtt{f}\}, \mathcal{F}_1 = \{\mathtt{i}\}$, and $\mathcal{F}_0 = \{\mathtt{e}\}$.

We recall that, if $\Sigma = (\mathcal{P}, \mathcal{F})$ is a signature, a $\Sigma$-structure is a tuple $S = (D, Pred, Fun)$, where:

- $D$ is a non-empty set called the *domain* of the structure;
- for each each predicate symbol $P \in \mathcal{P}$ there is a predicate $P^S \in Pred$ having the same arity as $P$;
- for each predicate symbol $f \in \mathcal{F}$ there is a predicate $f^S \in Pred$ is a function having the same arity as $f$.

**Example 29.** *Here are some examples of $\Sigma$-structures:*

1. $S_1 = (\mathbb{Z}, \{=\}, \{+, -, 0\})$;

2. $S_2 = (\mathbb{R}^*, \{=\}, \{\times, \cdot^{-1}, 1\})$;
3. $S_3 = (\mathbb{N}, \{=\}, \{+, s, 0\})$;
4. $S_4 = (\mathbb{N}, \{<\}, \{+, s, 0\})$.
5. $S_5 = (\mathbb{Z}, \{<\}, \{+, -, 0\})$.

*The structure $S_1$ has the domain $\mathbb{Z}$ (the set of integers), the predicate associated to the predicate symbol* P *is $=$ (the equality predicate for integers), the function $+$ is the addition function for integers and it is associated to the function symbol* f*, $-$ is the unary minus function associated to the function symbol* i*, and the constant symbol* e *has 0 as the associated constant.*

*The structure $S_2$ has the domain $\mathbb{R}^*$ (the set of positive reals), the predicate associated to the predicate symbol* P *is $=$ (the equality predicate over positive reals), the function $\times$ is the multiplication function over positive reals associated to the function symbol* f*, $\cdot^{-1}$ is the unary function associated to the function symbol* i *that computes the inverse of a positive real number (e.g.* $5^{-1} = \frac{1}{5}$ *, and* $\left(\frac{1}{10}\right)^{-1} = 10$*), and the constant 1 is associated to the constant symbol* e*.*

*The structure $S_3$ has the domain $\mathbb{N}$ (the set of naturals), the predicate associated to the predicate symbol* P *is $=$ (the equality predicate for natural numbers), the function $+$ is the addition function for naturals associated to the function symbol* f*, $s$ is the* successor *function (which associates to a natural number the next natural number – e.g., $s(7) = 8$) associated to the function symbol* i*, and the constant 0 is associated to the constant symbol* e*.*

*The structure $S_4$ has the domain $\mathbb{N}$ (the set of naturals), the predicate associated to the symbol* P *is $<$ (the* smaller than *relation over naturals), the function $+$ is the addition function for naturals associated to the function symbol* f*, $s$ is the successor function associated to the function symbol* i*, and the constant 0 is associated to the constant symbol* e*.*

*The structure $S_5$ is similar to $S_1$, but the predicate symbol $P$ is associated with the* less than *relation instead of* equality*.*

*Using the notation above, we have that* $\mathtt{P}^{S_4} = <$*,* $\mathtt{f}^{S_2} = \times$*, and* $\mathtt{e}^{S_1} = 0$*.*

## 4.1 Assignments

As in the propositional logic, in order to obtain the truth value of a formula in a structure, we have to start by fixing some concrete values to the syntactic symbols over which the formula is built. In the case of first order logic, we start with the variables.

**Definition 30** (Assignment)**.** *Let $\Sigma$ be a signature and let $S$ be $\Sigma$-structure with the domain $D$.*

*An $S$-assignment is any function*

$$\alpha : \mathcal{X} \to D.$$

**Example 31.** *Consider the $S_1$-assignment $\alpha_1 : \mathcal{X} \to \mathbb{Z}$ defined as follows:*

1. $\alpha_1(\mathsf{x}_1) = 5$;
2. $\alpha_1(\mathsf{x}_2) = 5$;
3. $\alpha_1(\mathsf{x}_3) = 6$;
4. $\alpha_1(x) = 0$ *for all* $x \in \mathcal{X} \setminus \{\mathsf{x}_1, \mathsf{x}_2, \mathsf{x}_3\}$.

**Example 32.** *Consider the $S_1$-assignment $\alpha_2 : \mathcal{X} \to \mathbb{Z}$ defined as follows:*

1. $\alpha_2(\mathsf{x}_1) = 6$;
2. $\alpha_2(\mathsf{x}_2) = 5$;
3. $\alpha_2(\mathsf{x}_3) = 6$;
4. $\alpha_2(x) = 0$ *for all* $x \in \mathcal{X} \setminus \{\mathsf{x}_1, \mathsf{x}_2, \mathsf{x}_3\}$.

In the presence of an assignment $\alpha$ we can compute the value of a term in the assignment $\alpha$. For this, we use an extension of $\alpha$, denoted $\overline{\alpha}$,

$$\overline{\alpha} : \mathcal{T} \to D,$$

which is defined below.

**Definition 33** (The Value of a Term in an Assignment)**.** *Given an $S$-assignment $\alpha$ and a term $t \in \mathcal{T}$ over the signature $\Sigma$,* the value of the term $t$ in the assignment $\alpha$ *is an element of the domain $D$, denoted by $\overline{\alpha}(t)$, and computed recursively as follows:*

1. $\overline{\alpha}(c) = c^S$ *if $c \in \mathcal{F}_0$ is a constant symbol;*
2. $\overline{\alpha}(x) = \alpha(x)$ *if $x \in \mathcal{X}$ is a variable;*
3. $\overline{\alpha}(f(t_1, \ldots, t_n)) = f^S(\overline{\alpha}(t_1), \ldots, \overline{\alpha}(t_n))$ *if $f \in \mathcal{F}_n$ is a function symbol of arity $n$, and $t_1, \ldots, t_n$ are terms.*

**Example 34.** *Continuing Example 31, where $\alpha_1$ is an $S_1$-assignment, we have that*

$$\begin{aligned}
\overline{\alpha_1}(\mathsf{f}(\mathsf{i}(\mathsf{x}_1), \mathsf{e})) &= \overline{\alpha_1}(\mathsf{i}(\mathsf{x}_1)) + \overline{\alpha_1}(\mathsf{e}) \\
&= -(\overline{\alpha_1}(\mathsf{x}_1)) + \mathsf{e}^{S_1} \\
&= -(\alpha_1(\mathsf{x}_1)) + 0 \\
&= -5 + 0 \\
&= -5.
\end{aligned}$$

*Therefore, the value of the term $\mathsf{f}(\mathsf{i}(\mathsf{x}_1), \mathsf{e})$ in the assignment $\alpha_1$ is $-5$.*

**Definition 35** (Updating an Assignment)**.** *Given an assignment $\alpha$, a variable $x \in \mathcal{X}$ and a value $u \in D$, we denote by $\alpha[x \mapsto u]$ a new assignment, obtained from $\alpha$ by updating the value of the variable $x$ to $u$, formally defined as follows:*

$$\alpha[x \mapsto u] : \mathcal{X} \to D, \text{ s.t.}$$

1. $(\alpha[x \mapsto u])(x) = u$;
2. $(\alpha[x \mapsto u])(y) = \alpha(y)$, *for any* $y \in \mathcal{X} \setminus \{x\}$.

**Example 36.** *For example, the assignment $\alpha_1[\mathsf{x}_1 \mapsto 6]$ is exactly the same as the assignment $\alpha_2$ defined earlier. The value of the term $\mathsf{f}(\mathsf{i}(\mathsf{x}_1), \mathsf{e})$ in the assignment $\alpha_1[\mathsf{x}_1 \mapsto 6]$ is $\overline{\alpha_1[\mathsf{x}_1 \mapsto 6]}(\mathsf{f}(\mathsf{i}(\mathsf{x}_1), \mathsf{e})) = -6$.*

**Exercise 37.** *Compute the values below:*

1. $\overline{\alpha_1[\mathsf{x}_1 \mapsto 10]}(\mathsf{f}(\mathsf{i}(\mathsf{x}_1), \mathsf{e}))$;
2. $\overline{\alpha_1[\mathsf{x}_2 \mapsto 10]}(\mathsf{f}(\mathsf{i}(\mathsf{x}_1), \mathsf{e}))$;
3. $\overline{\alpha_1[\mathsf{x}_2 \mapsto 10][\mathsf{x}_1 \mapsto 10]}(\mathsf{f}(\mathsf{i}(\mathsf{x}_1), \mathsf{e}))$.

## 4.2 The Truth Value of a First-Order Logic Formula

Let $\Sigma$ be a signature, let $S$ be a $\Sigma$-structure and let $\alpha$ be an $S$-assignment. Having fixed the elements above, we may compute the truth value of a first-order formula constructed over the signature $\Sigma$.

By writing $S, \alpha \models \varphi$ we denote the fact that the formula $\varphi$ is true in the structure $S$ with the assignment $\alpha$. By writing $S, \alpha \not\models \varphi$ we denote the fact

that the formula $\varphi$ is not true (is false) in the structure $S$ with the assignment $\alpha$.

The notation $S, \alpha \models \varphi$ can also be read as *$S$ satisfies $\varphi$ with the assignment $\alpha$*, and $S, \alpha \not\models \varphi$ can also be read as *$S$ does not satisfy $\varphi$ with the assignment $\alpha$*.

**Definition 38.** *The fact that a structure $S$ satisfies a formula $\varphi$ with an assignment $\alpha$ (or equivalently, that $\varphi$ is true in the structure $S$ with the assignment $\alpha$) is defined inductively as follows:*

1. $S, \alpha \models P(t_1, \ldots, t_n)$ *iff* $P^S(\overline{\alpha}(t_1), \ldots, \overline{\alpha}(t_n))$;
2. $S, \alpha \models \neg\varphi$ *iff* $S, \alpha \not\models \varphi$;
3. $S, \alpha \models (\varphi_1 \wedge \varphi_2)$ *iff* $S, \alpha \models \varphi_1$ *and* $S, \alpha \models \varphi_2$;
4. $S, \alpha \models (\varphi_1 \vee \varphi_2)$ *iff* $S, \alpha \models \varphi_1$ *or* $S, \alpha \models \varphi_2$;
5. $S, \alpha \models (\varphi_1 \to \varphi_2)$ *iff* $S, \alpha \not\models \varphi_1$ *or* $S, \alpha \models \varphi_2$;
6. $S, \alpha \models (\varphi_1 \leftrightarrow \varphi_2)$ *iff (1) both* $S, \alpha \models \varphi_1$ *and* $S, \alpha \models \varphi_2$, *or (2)* $S, \alpha \not\models \varphi_1$ *and* $S, \alpha \not\models \varphi_2$;
7. $S, \alpha \models (\exists x.\varphi)$ *iff there is* $u \in D$ *such that* $S, \alpha[x \mapsto u] \models \varphi$;
8. $S, \alpha \models (\forall x.\varphi)$ *iff for all* $u \in D$, *we have that* $S, \alpha[x \mapsto u] \models \varphi$.

**Example 39.** *We shall work over the signature* $\Sigma = (\{\mathtt{P}\}, \{\mathtt{f}, \mathtt{i}, \mathtt{e}\})$, *the* $\Sigma$*-structure* $S_1 = (\mathbb{Z}, \{=\}, \{+, -, 0\})$ *and the* $S_1$*-assignments* $\alpha_1, \alpha_2$.

*We have that*

$$\begin{aligned} S_1, \alpha_1 \models \mathtt{P}(\mathtt{x}_1, \mathtt{x}_1) \text{ iff } & \quad \mathtt{P}^{S_1}(\overline{\alpha_1}(\mathtt{x}_1), \overline{\alpha_1}(\mathtt{x}_1)) \\ \text{iff } & \quad \overline{\alpha_1}(\mathtt{x}_1) = \overline{\alpha_1}(\mathtt{x}_1) \\ \text{iff } & \quad \alpha_1(\mathtt{x}_1) = \alpha_1(\mathtt{x}_1) \\ \text{iff } & \quad 5 = 5. \end{aligned}$$

*As* $5 = 5$, *we have that* $S_1, \alpha_1 \models \mathtt{P}(\mathtt{x}_1, \mathtt{x}_1)$, *meaning that the formula* $\mathtt{P}(\mathtt{x}_1, \mathtt{x}_1)$ *is true in the structure* $S_1$ *with the assignment* $\alpha_1$. *Equivalently, we say that* $S_1$ *satisfies* $\mathtt{P}(\mathtt{x}_1, \mathtt{x}_1)$ *with the assignment* $\alpha_1$.

**Example 40.** *Continuing the previous example, we have that*

$$\begin{aligned} S_1, \alpha_1 \models \mathtt{P}(\mathtt{x}_1, \mathtt{x}_3) \text{ iff } & \quad P^{S_1}(\overline{\alpha_1}(\mathtt{x}_1), \overline{\alpha_1}(\mathtt{x}_3)) \\ \text{iff } & \quad \overline{\alpha_1}(\mathtt{x}_1) = \overline{\alpha_1}(\mathtt{x}_3) \\ \text{iff } & \quad \alpha_1(\mathtt{x}_1) = \alpha_1(\mathtt{x}_3) \\ \text{iff } & \quad 5 = 6. \end{aligned}$$

*As* $5 \neq 6$, *we have that* $S_1, \alpha_1 \not\models \mathtt{P}(\mathtt{x}_1, \mathtt{x}_3)$, *meaning that* $\mathtt{P}(\mathtt{x}_1, \mathtt{x}_3)$ *is false in the structure* $S_1$ *with the assignment* $\alpha_1$. *Equivalently, we say that* $S_1$ *does not satisfy* $\mathtt{P}(\mathtt{x}_1, \mathtt{x}_3)$ *with the assignment* $\alpha_1$.

**Example 41.** *Continuing the previous example, we have that*

$$\begin{aligned} S_1, \alpha_1 \models \neg \mathtt{P}(\mathtt{x}_1, \mathtt{x}_3) \text{ iff } & \quad S_1, \alpha_1 \not\models \mathtt{P}(\mathtt{x}_1, \mathtt{x}_3) \\ \text{iff } & \quad \text{not } P^{S_1}(\overline{\alpha_1}(\mathtt{x}_1), \overline{\alpha_1}(\mathtt{x}_3)) \\ \text{iff } & \quad \text{not } \overline{\alpha_1}(\mathtt{x}_1) = \overline{\alpha_1}(\mathtt{x}_3) \\ \text{iff } & \quad \overline{\alpha_1}(\mathtt{x}_1) \neq \overline{\alpha_1}(\mathtt{x}_3) \\ \text{iff } & \quad \alpha_1(\mathtt{x}_1) \neq \alpha_1(\mathtt{x}_3) \\ \text{iff } & \quad 5 \neq 6. \end{aligned}$$

*As* $5 \neq 6$, *we have that* $S_1, \alpha_1 \models \neg \mathtt{P}(\mathtt{x}_1, \mathtt{x}_3)$, *meaning that the formula* $\neg \mathtt{P}(\mathtt{x}_1, \mathtt{x}_3)$ *is true in the structure* $S_1$ *with the assignment* $\alpha_1$. *Equivalently, we say that* $S_1$ *satisfies* $\neg \mathtt{P}(\mathtt{x}_1, \mathtt{x}_3)$ *with the assignment* $\alpha_1$.

**Example 42.** *Continuing the previous example, we have that*

$$\begin{aligned} & S_1, \alpha_1 \models \mathtt{P}(\mathtt{x}_1, \mathtt{x}_1) \wedge \neg \mathtt{P}(\mathtt{x}_1, \mathtt{x}_3) && \text{iff} \\ & S_1, \alpha_1 \models \mathtt{P}(\mathtt{x}_1, \mathtt{x}_1) \text{ și } S_1, \alpha_1 \models \neg \mathtt{P}(\mathtt{x}_1, \mathtt{x}_3) && \text{iff} \\ & \ldots \text{ and } \ldots && \text{iff} \\ & 5 = 5 \text{ and } 5 \neq 6. \end{aligned}$$

*As* $5 = 5$ *și* $5 \neq 6$, *we have that* $S_1, \alpha_1 \models \mathtt{P}(\mathtt{x}_1, \mathtt{x}_1) \wedge \neg \mathtt{P}(\mathtt{x}_1, \mathtt{x}_3)$.

**Example 43.** *Continuing the previous example, we have that*

$$S_1, \alpha_1 \models \mathtt{P}(\mathtt{x}_1, \mathtt{x}_3) \vee \mathtt{P}(\mathtt{x}_1, \mathtt{x}_1) \text{ iff } S_1, \alpha_1 \models \mathtt{P}(\mathtt{x}_1, \mathtt{x}_3) \text{ or } S_1, \alpha_1 \models \mathtt{P}(\mathtt{x}_1, \mathtt{x}_1).$$

*We have already seen that* $S_1, \alpha_1 \models \mathtt{P}(\mathtt{x}_1, \mathtt{x}_3)$, *so* $S_1, \alpha_1 \models \mathtt{P}(\mathtt{x}_1, \mathtt{x}_3) \vee \mathtt{P}(\mathtt{x}_1, \mathtt{x}_1)$ *(even if* $S_1, \alpha_1 \not\models \mathtt{P}(\mathtt{x}_1, \mathtt{x}_1)$*).*

**Example 44.** *Continuing the previous example, we have that*

$$\begin{aligned} & S_1, \alpha_1 \models \exists \mathtt{x}_1 . \mathtt{P}(\mathtt{x}_1, \mathtt{x}_3) && \text{iff} \\ & \text{exists } u \in D \text{ a.î. } S_1, \alpha_1[\mathtt{x}_1 \mapsto u] \models \mathtt{P}(\mathtt{x}_1, \mathtt{x}_3) && \text{iff} \\ & \text{exists } u \in D \text{ a.î. } P^{S_1}(\overline{\alpha_1[\mathtt{x}_1 \mapsto u]}(\mathtt{x}_1), \overline{\alpha_1[\mathtt{x}_1 \mapsto u]}(\mathtt{x}_3)) && \text{iff} \\ & \text{exists } u \in D \text{ a.î. } \overline{\alpha_1[\mathtt{x}_1 \mapsto u]}(\mathtt{x}_1) = \overline{\alpha_1[\mathtt{x}_1 \mapsto u]}(\mathtt{x}_3) && \text{iff} \\ & \text{exists } u \in D \text{ a.î. } \alpha_1[\mathtt{x}_1 \mapsto u](\mathtt{x}_1) = \alpha_1[\mathtt{x}_1 \mapsto u](\mathtt{x}_3) && \text{iff} \\ & \text{exists } u \in D \text{ a.î. } u = \alpha_1(\mathtt{x}_3) && \text{iff} \\ & \text{exists } u \in D \text{ a.î. } u = 6. \end{aligned}$$

*Since there exists $u$ (namely $u = 6$) so that $u = 6$, we have that* $S_1, \alpha_1 \models \exists x_1.\mathtt{P}(x_1, x_3)$.

**Example 45.** *Continuing the previous example, we have that*

$$
\begin{array}{ll}
S_1, \alpha_1 \models \forall x_1.\exists x_3.\mathtt{P}(x_1, x_3) & \textit{iff} \\
\textit{for all } u \in D, \textit{ we have } S_1, \alpha_1[x_1 \mapsto u] \models \exists x_3.\mathtt{P}(x_1, x_3) & \textit{iff} \\
\textit{for all } u \in D, \textit{ there is } v \in D \textit{ a.î. } S_1, \alpha_1[x_1 \mapsto u][x_3 \mapsto v] \models \mathtt{P}(x_1, x_3) & \textit{iff} \\
\ldots & \textit{iff} \\
\textit{for all } u \in D, \textit{ there is } v \in D \textit{ a.î. } u = v. &
\end{array}
$$

*As for any integer $u$, there exists an integer $v$ so that $u = v$, we have that* $S_1, \alpha_1 \models \forall x_1.\exists x_3.\mathtt{P}(x_1, x_3)$.

**Exercise 46.** *Show that* $S_1, \alpha_1 \models \forall x_1.\exists x_3.\mathtt{P}(x_1, \mathtt{i}(x_3))$.

## 4.3 Satisfiability in a Fixed Structure

**Definition 47** (Satisfiability in a Fixed Structure)**.** *A formula $\varphi$* is satisfiable in a structure *$S$ if there exists an $S$-assignment $\alpha$ such that*

$$S, \alpha \models \varphi.$$

**Example 48.** *The formula $\mathtt{P}(x_1, x_3)$ is satisfiable in the structure $S_1$ because there is an assignment, for instance $\alpha_2$, such that $S_1, \alpha_2 \models \mathtt{P}(x_1, x_3)$.*

**Exercise 49.** *Show that $\neg\mathtt{P}(x_1, x_1)$ is not satisfiable in $S_1$ (because, for each assignment $\alpha$, we have $S_1, \alpha \not\models \neg\mathtt{P}(x_1, x_1)$).*

## 4.4 Validity in a Fixed Structure

**Definition 50** (Validity in a Fixed Structure)**.** *A formula $\varphi$* is valid in a structure *$S$ if, for any $S$-assignment $\alpha$, we have that*

$$S, \alpha \models \varphi.$$

**Example 51.** *The formula $\mathtt{P}(x_1, x_3)$ is not valid in the structure $S_1$, because there exists an assignment, namely $\alpha_1$, having the property that $S_1, \alpha_1 \not\models \mathtt{P}(x_1, x_3)$.*

**Exercise 52.** *Show that $\mathtt{P}(x_1, x_1)$ is valid in the structure $S_1$ (because, for any assignment $\alpha$, $S_1, \alpha \models \mathtt{P}(x_1, x_1)$).*

## 4.5 Satisfiability

**Definition 53** (Satisfiability)**.** *A formula* $\varphi$ is satisfiable *if there exists a structure* $S$ *and an* $S$*-assignment* $\alpha$ *such that*

$$S, \alpha \models \varphi.$$

**Example 54.** *The formula* $\neg \mathtt{P}(\mathtt{x}_1, \mathtt{x}_1)$ *is satisfiable, because there exists a structure (namely* $S_5$*) and an* $S_5$*-assignment (namely* $\alpha_1$*) so that* $S_5, \alpha_1 \models \neg \mathtt{P}(\mathtt{x}_1, \mathtt{x}_1)$ *(because* $5 \not< 5$*).*

*Remark. Because* $S_5$ *and* $S_1$ *have the same domain, the assignment* $\alpha_1$ *is both an* $S_1$*-assignment as well as an* $S_5$*-assignment.*

**Remark.** *A formula could be unsatisfiable in a fixed structure (for example* $\neg \mathtt{P}(\mathtt{x}_1, \mathtt{x}_1)$ *is not satisfiable in the structure* $S_1$*) but it could still be satisfiable (for example* $\neg \mathtt{P}(\mathtt{x}_1, \mathtt{x}_1)$*, as we have seen above).*

## 4.6 Validity

**Definition 55** (Validity)**.** *A formula* $\varphi$ is valid *if, for any structure* $S$ *and for any* $S$*-assignment* $\alpha$*, we have that*

$$S, \alpha \models \varphi.$$

**Example 56.** *The formula* $\mathtt{P}(\mathtt{x}_1, \mathtt{x}_1)$ *is not valid, because* $S_5, \alpha_1 \not\models \mathtt{P}(\mathtt{x}_1, \mathtt{x}_1)$*.*

*The formula* $\mathtt{P}(\mathtt{x}_1, \mathtt{x}_1) \to \mathtt{P}(\mathtt{x}_1, \mathtt{x}_1)$ *is valid.*

**Remark.** *A formula could be valid in a fixed structure (for example* $\mathtt{P}(\mathtt{x}_1, \mathtt{x}_1)$ *is valid in the structure* $S_1$*) and still not be valid (for example,* $\mathtt{P}(\mathtt{x}_1, \mathtt{x}_1)$ *is not valid because* $S_5, \alpha_1 \not\models \mathtt{P}(\mathtt{x}_1, \mathtt{x}_1)$ *).*

## 4.7 Semantical Consequence

**Definition 57.** *A formula* $\varphi$ is a semantical (or logical) consequence of the formulae $\varphi_1, \ldots, \varphi_n$ in a fixed structure $S$*, denoted by* $\varphi_1, \ldots, \varphi_n \models_S \varphi$*, if, for any* $S$*-assignment* $\alpha$ *for which* $S, \alpha \models \varphi_1$*,* $S, \alpha \models \varphi_2$*, . . . ,* $S, \alpha \models \varphi_n$*, we also have that* $S, \alpha \models \varphi$*.*

**Example 58.** *We have* $\mathtt{P}(\mathtt{x}, \mathtt{y}) \models_{S_1} \mathtt{P}(\mathtt{y}, \mathtt{x})$*, because, for any* $S_1$*-assignment* $\alpha$ *with the property that* $S_1, \alpha \models \mathtt{P}(\mathtt{x}, \mathtt{y})$ *(meaning that* $\alpha(\mathtt{x}) = \alpha(\mathtt{y})$*), we also have* $S_1, \alpha \models \mathtt{P}(\mathtt{y}, \mathtt{x})$ *(meaning* $\alpha(\mathtt{y}) = \alpha(\mathtt{x})$*).*

*We have that* $\mathtt{P}(\mathtt{x}, \mathtt{y}) \not\models_{S_5} \mathtt{P}(\mathtt{y}, \mathtt{x})$*, because, for the assignment* $\alpha(\mathtt{x}) = 5$*,* $\alpha(\mathtt{y}) = 6$*, we have* $S_5, \alpha \models \mathtt{P}(\mathtt{x}, \mathtt{y})$ *(that is,* $5 < 6$*), bu* $S_5, \alpha \not\models \mathtt{P}(\mathtt{y}, \mathtt{x})$ *(*$6 \not< 5$*).*

**Definition 59.** *A formula $\varphi$ is a semantical (or logicl) consequence of the formulae $\varphi_1, \ldots, \varphi_n$, denoted by $\varphi_1, \ldots, \varphi_n \models \varphi$, if*

$$\varphi_1, \ldots, \varphi_n \models_S \varphi$$

*for any structure $S$.*

**Example 60.** *We have $\mathtt{P}(\mathtt{x}, \mathtt{y}) \not\models \mathtt{P}(\mathtt{y}, \mathtt{x})$, because there exists a structure (namely $S_5$) so that $\mathtt{P}(\mathtt{x}, \mathtt{y}) \not\models_{S_5} \mathtt{P}(\mathtt{y}, \mathtt{x})$.*

**Exercise 61.** *Show that*

$$\forall \mathtt{x}.\forall \mathtt{y}.\forall \mathtt{z}.(\mathtt{P}(\mathtt{x}, \mathtt{y}) \wedge \mathtt{P}(\mathtt{y}, \mathtt{z}) \rightarrow \mathtt{P}(\mathtt{x}, \mathtt{z})), \mathtt{P}(\mathtt{x}_1, \mathtt{x}_2), \mathtt{P}(\mathtt{x}_2, \mathtt{x}_3) \models \mathtt{P}(\mathtt{x}_1, \mathtt{x}_3).$$

Of course, in the previous notations (as in the propositional logic), the list $\varphi_1, \varphi_2 ... \varphi_n$ denotes the set having as elements the enumerated formulae.

## 4.8 Consistent set of formulae

**Definition 62.** *A set of formulae* $\Gamma$ is consistent in a structure $S$ *if, by definition,*

$$\textit{there exists } \alpha \textit{ s.t. } S, \alpha \models \varphi \textit{ for any } \varphi \in \Gamma.$$

**Example 63.** *We have that $\{\mathtt{P}(\mathtt{x}, \mathtt{y}), \mathtt{P}(\mathtt{y}, \mathtt{x})\}$ is consistent in $S_1$, but not in $S_4$.*

**Definition 64.** *A set of formulae* $\Gamma$ is consistent *if, by definition, there exists a structure in which $\Gamma$ is consistent.*

**Example 65.** *We have that $\{\mathtt{P}(\mathtt{x}, \mathtt{y}), \mathtt{P}(\mathtt{y}, \mathtt{x})\}$ is consistent (because, for example, it is consistent in $S_1$).*

**Example 66.** *We have that $\{\mathtt{P}(\mathtt{x}, \mathtt{y}), \neg\mathtt{P}(\mathtt{x}, \mathtt{y})\}$ is not consistent.*

## 4.9 Exercises

We recall here the structures show in Example 29:

1. $S_1 = (\mathbb{Z}, \{=\}, \{+, -, 0\})$;
2. $S_2 = (\mathbb{R}^*, \{=\}, \{\times, \cdot^{-1}, 1\})$;
3. $S_3 = (\mathbb{N}, \{=\}, \{+, s, 0\})$;
4. $S_4 = (\mathbb{N}, \{<\}, \{+, s, 0\})$.

5. $S_5 = (\mathbb{Z}, \{<\}, \{+, -, 0\})$.

These structures will be used in the exercises below.

**Exercise 67.** *Establish whether the following sentences hold:*

1. $S_1, \alpha_1 \models \mathtt{P}(\mathtt{x}_2, \mathtt{x}_3)$;
2. $S_1, \alpha_1 \models \neg\mathtt{P}(\mathtt{x}_2, \mathtt{x}_3)$;
3. $S_1, \alpha_1 \models \neg\mathtt{P}(\mathtt{x}_2, \mathtt{x}_3) \wedge \mathtt{P}(\mathtt{x}_1, \mathtt{x}_1)$;
4. $S_1, \alpha_1 \models \exists \mathtt{x}_3.\mathtt{P}(\mathtt{x}_2, \mathtt{x}_3)$;
5. $S_1, \alpha_1 \models \forall \mathtt{x}_2.\exists \mathtt{x}_3.\mathtt{P}(\mathtt{x}_2, \mathtt{x}_3)$;
6. $S_1, \alpha_1 \models \exists \mathtt{x}_3.\forall \mathtt{x}_2.\mathtt{P}(\mathtt{x}_2, \mathtt{x}_3)$;
7. $S_1, \alpha_2 \models \forall \mathtt{x}_2.\exists \mathtt{x}_3.\mathtt{P}(\mathtt{x}_2, \mathtt{i}(\mathtt{x}_3))$;

**Exercise 68.** *Find for each item below a $S_2$-assignment $\alpha_3$ such that:*

1. $S_2, \alpha_3 \models \mathtt{P}(\mathtt{x}_1, \mathtt{x}_2)$;
2. $S_2, \alpha_3 \models \mathtt{P}(\mathtt{f}(\mathtt{x}_1, \mathtt{x}_2), \mathtt{x}_3)$;
3. $S_2, \alpha_3 \models \mathtt{P}(\mathtt{f}(\mathtt{x}_1, \mathtt{x}_2), \mathtt{i}(\mathtt{x}_3))$;
4. $S_2, \alpha_3 \models \mathtt{P}(\mathtt{x}, \mathtt{e})$;
5. $S_2, \alpha_3 \models \exists \mathtt{y}.\mathtt{P}(\mathtt{x}, \mathtt{i}(\mathtt{y}))$;
6. $S_2, \alpha_3 \models \forall \mathtt{y}.\exists \mathtt{x}.\mathtt{P}(\mathtt{x}, \mathtt{i}(\mathtt{y}))$.

**Exercise 69.** *Show that the formulae below are valid in $S_2$:*

1. $\forall \mathtt{x}.\exists \mathtt{y}.\mathtt{P}(\mathtt{x}, \mathtt{i}(\mathtt{y}))$;
2. $\forall \mathtt{x}.\mathtt{P}(\mathtt{f}(\mathtt{x}, \mathtt{e}), \mathtt{x})$;
3. $\forall \mathtt{x}.\mathtt{P}(\mathtt{x}, \mathtt{i}(\mathtt{i}(\mathtt{x})))$.

**Exercise 70.** *Show that $\forall \mathtt{x}.\exists \mathtt{y}.\mathtt{P}(\mathtt{x}, \mathtt{i}(\mathtt{y}))$ is not valid in $S_3$.*

**Exercise 71.** *Find a formula that is satisfiable in $S_1$ but not satisfiable in $S_3$.*

**Exercise 72.** *Show that $\forall \mathtt{x}.\exists \mathtt{y}.\mathtt{P}(\mathtt{x}, \mathtt{y})$ is not valid.*

**Exercise 73.** *Show that $(\forall \mathtt{x}.\mathtt{P}(\mathtt{x}, \mathtt{x})) \rightarrow \exists \mathtt{x}_2.\mathtt{P}(\mathtt{x}_1, \mathtt{x}_2)$ is valid.*

**Exercise 74.** *Show that $\forall \mathtt{x}.\exists \mathtt{y}.\mathtt{P}(\mathtt{y}, \mathtt{x})$ is not valid.*

**Exercise 75.** *Show that $\forall \mathtt{x}.\neg\mathtt{P}(\mathtt{x}, \mathtt{x})$ is satisfiable.*

**Exercise 76.** *Show that $\forall \mathtt{x}.\neg\mathtt{P}(\mathtt{x}, \mathtt{x}) \wedge \exists \mathtt{x}.\mathtt{P}(\mathtt{x}, \mathtt{x})$ is not satisfiable.*