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 τ?

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 φ above true? The formula states that between any two values of the variables x, y there is a third value, of the variable 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 Σ *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 (Σ-structure)**.** *If* $\Sigma = (\mathcal{P}, \mathcal{F})$ *is a signature, a* Σ-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* Σ*-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 Σ we can associate many structures, which are called Σ-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).*

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 Σ. In propositional logic, there was just one language, $\mathbb{PL}$.

Next, we shall fix a signature Σ 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 5. *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* Σ*-terms, because the definition of* $\mathcal{T}$ *depends on* Σ*.*

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

Example 6. *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:* $\mathtt{a}$*,* $\mathtt{b}$*,* $\mathtt{x}$*,* $\mathtt{y}$*,* $\mathtt{x}_1$*,* $\mathtt{y}'$*,* $\mathtt{i(a)}$*,* $\mathtt{i(x)}$*,* $\mathtt{i(i(a))}$*,* $\mathtt{i(i(x))}$*,* $\mathtt{f(a,b)}$*,* $\mathtt{i(f(a,b))}$*,* $\mathtt{f(f(x,a),f(y,y))}$*.*

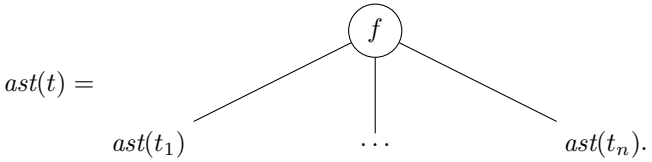
Exercise 7. *Identify the* Σ*-terms in the following list :*

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. $\mathtt{a(i(x))}$*.*

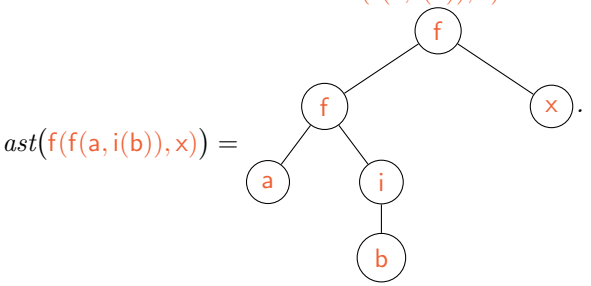
We denote terms by the letters t, s, t_1, t_2, s_1, t', etc. Even if terms are usually written as a string of characters, they have an associated abstract syntax tree defined as follows:

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

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 8. *Compute the abstract syntax trees for all terms in Example 6.*

3.3 Atomic formulae

Definition 9 (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 10. *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 11. *Explain why* $\mathsf{P(a)}$*,* P*,* $\mathsf{i(i(x))}$ *are not atomic formulae over the signature in Example 10.*

3.4 First-Order Formulae

Definition 12 (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) $(\exists x.\varphi) \in \mathbb{FOL}$*.*

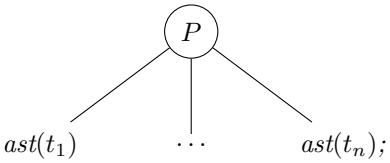
Remark. *In Definition 12, we find the logical connectives* $\neg, \wedge, \vee, \to$ *and* $\leftrightarrow$ *from propositional logic. The predicate symbols of arity* 0 *play the role of propositional variables (for now, at the syntactic level). The constructions* $(\forall x.\varphi)$ *and* $\exists x.\varphi)$ *are new.*

Example 13. *Recall* $\Sigma = (\{\mathtt{P}, \mathtt{Q}\}, \{\mathtt{f}, \mathtt{i}, \mathtt{a}, \mathtt{b}\})$ *from Example 6, 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:

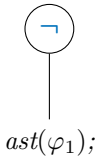
1. $\mathtt{P}(\mathtt{a},\mathtt{b})$*;*
2. $\mathtt{Q}(\mathtt{a},\mathtt{b})$*;*
3. $\mathtt{P}(\mathtt{a},\mathtt{x})$*;*
4. $\neg\mathtt{P}(\mathtt{a},\mathtt{b})$*;*
5. $(\mathtt{P}(\mathtt{a},\mathtt{b}) \wedge \neg\mathtt{Q}(\mathtt{a},\mathtt{b}))$*;*
6. $(\mathtt{P}(\mathtt{a},\mathtt{b}) \vee \neg\mathtt{Q}(\mathtt{x},\mathtt{y}))$*;*
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 14 (The Abstract Syntax Tree of formulae in $\mathbb{FOL}$)**.** *Formulae have an associated abstract syntax tree defined as follows:*

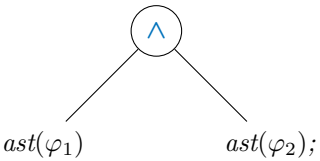
1. *if* $\varphi = P(t_1, \ldots, t_n)$, *then* $ast(\varphi) =$



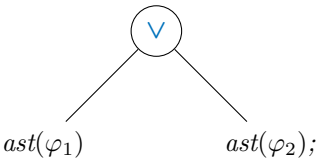
2. *if* $\varphi = \neg\varphi_1$, *then* $ast(\varphi) =$



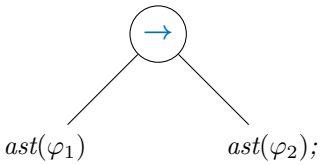
3. *if* $\varphi = (\varphi_1 \wedge \varphi_2)$, *then* $ast(\varphi) =$



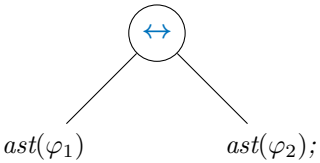
4. *if* $\varphi = (\varphi_1 \vee \varphi_2)$, *then* $ast(\varphi) =$



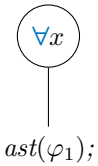
5. *if* $\varphi = (\varphi_1 \rightarrow \varphi_2)$, *then* $ast(\varphi) =$

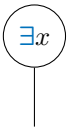


6. *if* $\varphi = (\varphi_1 \leftrightarrow \varphi_2)$, *then* $ast(\varphi) =$



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



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

Exercise 15. *Compute the abstract syntax tree of formulas shown in Example 13.*

3.5 The Brackets

The brackets (and) are used to mark the order of carrying out the logical operations (and, or, not, etc.). Next, we will drop certain extra brackets, just like in the case of propositional logic: if a formula can be interpreted as an abstract syntax in two or more ways, we will use brackets to fix the desired tree.

For example, $\varphi_1 \vee \varphi_2 \wedge \varphi_3$ could be understood as $((\varphi_1 \vee \varphi_2) \wedge \varphi_3)$ or as $(\varphi_1 \vee (\varphi_2 \wedge \varphi_3))$. In order to save brackets, we establish the following priority order of logical connectives:

$$\neg, \wedge, \vee, \rightarrow, \leftrightarrow, \forall, \exists,$$

where $\neg$ has the highest priority and $\exists$ has the lowest priority. When we are not 100% sure, it is better to use extra brackets.

Because of the order of priority for logical connectives, $\varphi_1 \vee \varphi_2 \wedge \varphi_3$ shall always be understood as $(\varphi_1 \vee (\varphi_2 \wedge \varphi_3))$ (because $\wedge$ has priority over $\vee$). As an anology, it works the same way as in arithmetic: $1 + 2 * 3$ will be understood as $1 + (2 \times 3)$, because $\times$ has priority over $+$ ($\times$ is similar to $\wedge$ and $+$ to $\vee$).

Exercise 16. *Write the formulas in Example 13 using the minimal number of brackets.*

In Section 3.12 we will discuss more about the interaction between the cuantifiers and the other logical connectives. We will see that we have some extra rules besides the above priorities.

3.6 Model english sentences as formulas 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

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

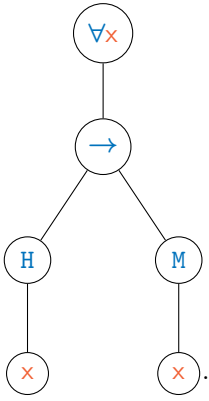
where $\mathtt{H}$ and $\mathtt{M}$ are predicate symbols of arity $ar(\mathtt{H}) = ar(\mathtt{M}) = 1$, and $\mathtt{s}$ is a functional symbol of arity $ar(\mathtt{s}) = 0$.

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 Σ-structures. An example of a Σ-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, $(\forall x.(H(x) \to M(x)))$, $H(s)$ and respectively $M(s)$, 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 $M(s)$ is necessarily true in *any* structure in which the formulae $(\forall x.(H(x) \to M(x)))$ and respectively $H(s)$ are true, not just in the structure S above.

3.7 Model arithmetic sentences as formulas 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 Σ:

1. $\Big(\forall x.\big(\forall y.\big(<(x, y) \to \exists z.\big(<(x, z) \wedge <(z, y)\big)\big)\big)\Big)$;
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. $=(+(x, y), z)$.

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. $\Big(\forall x.\big(\forall y.\big(x < y \to \exists z.\big(x < z \wedge z < y\big)\big)\big)\Big)$;
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 Σ-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 $\mathtt{x}, \mathtt{y}, \mathtt{z}$. Because the variables $\mathtt{x}, \mathtt{y}, \mathtt{z}$ 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 $\mathtt{x}, \mathtt{y}, \mathtt{z}$ that make the formula true (e.g. the values $1, 2, 3$ for $\mathtt{x}, \mathtt{y}$, and respectively $\mathtt{z}$).

3.8 The Variables of a Formula

By $vars(\varphi)$ we denote the variables of the formula φ. For example, we shall have $vars\big((\forall \mathtt{z}.(\mathtt{P}(\mathtt{x}, \mathtt{y})))\big) = \{\mathtt{x}, \mathtt{y}, \mathtt{z}\}$. We next define the function $vars : \mathbb{FOL} \to 2^{\mathcal{X}}$.

First of all, we define a function $vars : \mathcal{T} \to 2^{\mathcal{X}}$ as being the function that associates to a term (from the set $\mathcal{T}$) the set of variables occurring in that term. All following definitions are defined inductively, as the corresponding syntactic definitions are. We call them recursive definitions and we do not explicitly differentiate the base case and the inductive cases. We recall that the set $2^{\mathcal{X}}$ is the set of all subsets of $\mathcal{X}$.

Definition 17. *The function $vars : \mathcal{T} \to 2^{\mathcal{X}}$ is defined recursively as follows:*

1. $vars(c) = \emptyset$ *(if $c \in \mathcal{F}_0$ is a constant symbol);*
2. $vars(x) = \{x\}$ *(if $x \in \mathcal{X}$ is a variable);*
3. $vars(f(t_1, \ldots, t_n)) = \bigcup_{i \in \{1,\ldots,n\}} vars(t_i)$.

We can now define the homonymous (extended) function $vars : \mathbb{FOL} \to 2^{\mathcal{X}}$, which associates to any first-order formula the set of variables of the formula:

Definition 18. *The function $vars : \mathbb{FOL} \to 2^{\mathcal{X}}$ is define recursively as follows:*

1. $vars(P(t_1, \ldots, t_n)) = \bigcup_{i \in \{1,\ldots,n\}} vars(t_i)$;

2. $vars(\neg\varphi) = vars(\varphi)$;
3. $vars((\varphi_1 \wedge \varphi_2)) = vars(\varphi_1) \cup vars(\varphi_2)$;
4. $vars((\varphi_1 \vee \varphi_2)) = vars(\varphi_1) \cup vars(\varphi_2)$;
5. $vars((\varphi_1 \rightarrow \varphi_2)) = vars(\varphi_1) \cup vars(\varphi_2)$;
6. $vars((\varphi_1 \leftrightarrow \varphi_2)) = vars(\varphi_1) \cup vars(\varphi_2)$;
7. $vars\big((\exists x.\varphi)\big) = vars(\varphi) \cup \{x\}$;
8. $vars\big((\forall x.\varphi)\big) = vars(\varphi) \cup \{x\}$.

Observe that the variable x is added to the set of variables even if it appears only as next to the quantifier $\exists$ or $\forall$.

Example 19. *Consider the formula φ:*

$$\Big(\big(\forall \mathtt{x}.(\mathtt{P}(\mathtt{x},\mathtt{y}) \wedge \exists \mathtt{y}.(\mathtt{P}(\mathtt{z},\mathtt{f}(\mathtt{x},\mathtt{y})) \wedge \mathtt{P}(\mathtt{x},\mathtt{y})))\big) \wedge \mathtt{P}(\mathtt{x},\mathtt{x})\Big).$$

We have that $vars(\varphi) = \{\mathtt{x},\mathtt{y},\mathtt{z}\}$.

Exercise 20. *Consider $\Sigma = (\{\mathtt{P},\mathtt{Q}\},\{\mathtt{f},\mathtt{i},\mathtt{a},\mathtt{b}\})$, wher $ar(\mathtt{P}) = ar(\mathtt{Q}) = 2$, $ar(\mathtt{f}) = 2$, $ar(\mathtt{i}) = 1$, $ar(\mathtt{a}) = ar(\mathtt{b}) = 0$. Compute $vars(\varphi)$ for each formula φ in the list below:*

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

3.9 The Scope of a Quantifier - Analogy with Programming Languages

In a programming language, we may declare several variables having the same name. For example, in C, we could have the following code:

```
/*  1:*/ int f()
/*  2:*/ {
/*  3:*/   int s = 0;
/*  4:*/   for (int x = 1; x <= 10; ++x) {
/*  5:*/     for (int y = 1; y <= 10; ++y) {
/*  6:*/       s += x * y * z;
/*  7:*/       for (int x = 1; x <= 10; ++x) {
/*  8:*/         s += x * y * z;
/*  9:*/       }
/* 10:*/     }
/* 11:*/   }
/* 12:*/   return s;
/* 13:*/ }
```

In the code fragment above, there are three variables, two of them having the same name, `x`. The scope of the variable `x` declared on line 4 is between the lines $4-11$, and the scope of the variable `x` declared on line 7 is the lines $7-9$. This way, any occurence of the name `x` between lines $7-9$ refers to the second declaration of the variable, while any occurrence of the name `x` between lines $4-11$ (except lines $7-9$) refers to the second declaration of `x`. For example, the occurrence of `x` on line 6 refers to the variable x declared on line 4. The occurrence of `x` on line 8 refers to the variable x declared on line 7.

The lines $4-11$ represent the scope of the first declaration of `x`, and the lines $7-9$ represent the scope of the second declaration of `x`. The variable z is a global variable.

This is similar to first-order formulae. For example, in the formula:

$$\Big(\forall x.\big(\forall y.(P(x,y) \wedge P(x,z) \wedge (\exists x.P(x,y)))\big)\Big),$$

the variable x is quantified twice (the first time universally, the second time existentially). A quantification of a variable is called *binding*. A *binding* is similar, from the point of view of the scope of the variable, to defining a variable in a programming language.

This way, the scope D_1 of the variable x that is quantified universally is $\big(\forall y.(P(x,y) \wedge P(x,z) \wedge (\exists x.P(x,y)))\big)$, while the scope D_2 of the variable x that is quantified existentially is $P(x,y)$:

$$\Big(\forall \mathtt{x}.\ \underbrace{\big(\forall \mathtt{y}.\big(\mathtt{P}(\mathtt{x},\mathtt{y}) \wedge \mathtt{P}(\mathtt{x},\mathtt{z}) \wedge (\exists \mathtt{x}.\ \overbrace{\mathtt{P}(\mathtt{x},\mathtt{y})}^{\mathtt{D_2}})\big)\big)}_{\mathtt{D_1}}\Big)$$

The occurrences of a variable in the scope of a quantifier that binds the variable are called *bound occurrences*, while the occurrences of a variable outside of the scope of any quantifier that binds the variable are called *free occurrences*.

3.10 Free and Bound Occurrences of Variables

In this section we formally define the notion of free/bound occurrence and of free/bound variable. The free occurrences of a variable in first-order logic are, as an analogy, similar to global variables in a programming language.

Recall the definition of the *syntax abstract tree* associated to a first order formula and consider that the sentence *"on the path to the root"* has a formal definition.

Definition 21. *A* free occurrence of a variable x in a formula φ *is a node in the tree of the formula* φ *labeled by* x *and having the property that there is no node labeled* $\forall x$ *or* $\exists x$ *on the path to the root.*

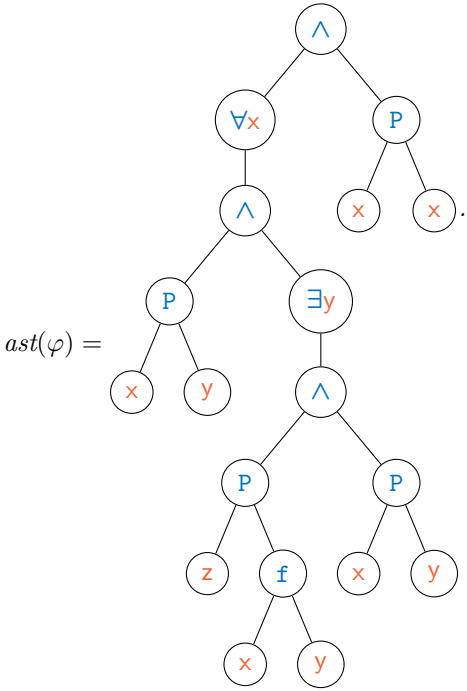
Definition 22. *A* bound occurrence of a variable x in a formula φ *is a node in the tree of the formula labeled by* x *and having the property that, on the path towards the root, there is at least a node labeled by* $\forall x$ *or by* $\exists x$*.*

The closest such node (labeled by $\forall x$ *or by* $\exists x$*) is the quantifier that* binds *that particular occurrence of the variable* x*.*

Example 23. *We next consider the formula*

$$\varphi = \Big(\Big(\forall \mathtt{x}.\big(\mathtt{P}(\mathtt{x},\mathtt{y}) \wedge \exists \mathtt{y}.\big(\mathtt{P}(\mathtt{z},\mathtt{f}(\mathtt{x},\mathtt{y})) \wedge \mathtt{P}(\mathtt{x},\mathtt{y})\big)\big)\Big) \wedge \mathtt{P}(\mathtt{x},\mathtt{x})\Big).$$

Its abstract syntax tree is:



In the formula φ above, the variable x has two free occurrences. The variable y has one free occurrence. The variable z has one free occurrence. All the free occurrences of the variables in the formula φ are underlined:

$$\varphi = \Big(\Big(\forall \mathtt{x}.\big(\mathtt{P}(\underline{\mathtt{x}},\underline{\mathtt{y}}) \wedge \exists \mathtt{y}.\big(\mathtt{P}(\underline{\mathtt{z}},\mathtt{f}(\underline{\mathtt{x}},\underline{\mathtt{y}})) \wedge \mathtt{P}(\underline{\mathtt{x}},\underline{\mathtt{y}})\big)\big)\Big) \wedge \mathtt{P}(\underline{\mathtt{x}},\underline{\mathtt{x}})\Big).$$

All the bound occurrences of the variables in the formula φ are underlined twice:

$$\varphi = \Big(\Big(\forall \mathtt{x}.\big(\mathtt{P}(\underline{\underline{\mathtt{x}}},\mathtt{y}) \wedge \exists \mathtt{y}.\big(\mathtt{P}(\mathtt{z},\mathtt{f}(\underline{\underline{\mathtt{x}}},\underline{\underline{\mathtt{y}}})) \wedge \mathtt{P}(\underline{\underline{\mathtt{x}}},\underline{\underline{\mathtt{y}}})\big)\big)\Big) \wedge \mathtt{P}(\mathtt{x},\mathtt{x})\Big).$$

Remark. *Some authors also consider that the nodes labeled by $\forall x$ or by $\exists x$ are bound occurrences of the variable x. In this course, we shall not consider the nodes $\forall x$ and respectively $\exists x$ as being occurrences of the variable x, but simply as binding sites for the variable x.*

3.11 Free Variables and Bound Variables

The set of variables that have at least one free occurrence in a formula φ is denoted $free(\varphi)$.

Definition 24. *The function* $free : \mathbb{FOL} \to 2^{\mathcal{X}}$ *is defined recursively as follows:*

1. $free(P(t_1, \ldots, t_n)) = vars(t_1) \cup \ldots \cup vars(t_n)$;
2. $free(\neg\varphi) = free(\varphi)$;
3. $free((\varphi_1 \wedge \varphi_2)) = free(\varphi_1) \cup free(\varphi_2)$;
4. $free((\varphi_1 \vee \varphi_2)) = free(\varphi_1) \cup free(\varphi_2)$;
5. $free((\varphi_1 \to \varphi_2)) = free(\varphi_1) \cup free(\varphi_2)$;
6. $free((\varphi_1 \leftrightarrow \varphi_2)) = free(\varphi_1) \cup free(\varphi_2)$;
7. $free\big((\forall x.\varphi)\big) = free(\varphi) \setminus \{x\}$;
8. $free\big((\exists x.\varphi)\big) = free(\varphi) \setminus \{x\}$.

Example 25. *Continuing the previous example, for the formula*

$$\varphi = \Big(\Big(\forall \mathtt{x}.\big(\mathtt{P}(\mathtt{x},\mathtt{y}) \wedge \exists \mathtt{y}.(\mathtt{P}(\mathtt{z},\mathtt{f}(\mathtt{x},\mathtt{y})) \wedge \mathtt{P}(\mathtt{x},\mathtt{y}))\big)\Big) \wedge \mathtt{P}(\mathtt{x},\mathtt{x})\Big),$$

we have that $free(\varphi) = \{\mathtt{x}, \mathtt{y}, \mathtt{z}\}$.

Exercise 26. *Compute* $vars(\varphi)$ *for each formula* φ *in Exercise 20.*

By $bound(\varphi)$ we denote the set of variables bound in a formula, that is the set of those variables x with the property that there exists in the formula at least one node labeled by $\forall x$ or by $\exists x$.

Definition 27. *The function* $bound : \mathbb{FOL} \to 2^{\mathcal{X}}$ *is defined recursively as follows:*

1. $bound(P(t_1, \ldots, t_n)) = \emptyset$;
2. $bound(\neg\varphi) = bound(\varphi)$;
3. $bound((\varphi_1 \wedge \varphi_2)) = bound(\varphi_1) \cup bound(\varphi_2)$;
4. $bound((\varphi_1 \vee \varphi_2)) = bound(\varphi_1) \cup bound(\varphi_2)$;

5. $bound((\varphi_1 \to \varphi_2)) = bound(\varphi_1) \cup bound(\varphi_2)$;
6. $bound((\varphi_1 \leftrightarrow \varphi_2)) = bound(\varphi_1) \cup bound(\varphi_2)$;
7. $bound((\forall x.\varphi)) = bound(\varphi) \cup \{x\}$;
8. $bound((\exists x.\varphi)) = bound(\varphi) \cup \{x\}$.

Exercise 28. *Compute $bound(\varphi)$ for each formula φ in Exercise 20.*

Exercise 29. *Compute $bound(\varphi)$, where*

$$\varphi = \Big(\Big(\forall \mathtt{x}.\big(\mathtt{P}(\mathtt{x},\mathtt{y}) \wedge \exists \mathtt{y}.\big(\mathtt{P}(\mathtt{z},\mathtt{f}(\mathtt{x},\mathtt{y})) \wedge \mathtt{P}(\mathtt{x},\mathtt{y})\big)\big)\Big) \wedge \mathtt{P}(\mathtt{x},\mathtt{x})\Big).$$

Definition 30 (Bound Variables of a Formula)**.** *The* bound variables of a formula φ *are the elements of* $bound(\varphi)$.

Definition 31 (Free Variables of a Formula)**.** *The* free variables of a formula φ *are the elements of the set* $free(\varphi)$.

Remark. *The sets $free(\varphi)$ and $bound(\varphi)$ could in general have common elements.*

Remark. *A variable can have several occurrences in a formula. Some of the occurrences could be free in the formula, while others could be bound.*

We must make the distinction between a free occurrence of a variable in a formula *and a* free variable of a formula*. The free occurrence is a node in the abstract syntax tree of the formula, while the free variable is an element of the set $\mathcal{X}$.*

We must also make the distinction between a bound occurrence of a variable in a formula *and a* bound variable of a formula*. The bound occurrence is a node in the abstract syntax tree, while the bound variable is an element of the set $\mathcal{X}$.*

3.12 The Scope of a Bound Variable and Brackets

Now that we have understood the scope of a bound variable, we may clarify a subtlety related to the order of priority of logical connectives. Up to this point, we have established that the order of priority is: $\neg, \wedge, \vee, \to, \leftrightarrow, \forall, \exists$. However, the quantifiers, $\forall$ and $\exists$, interact subtly with the other logical connectives, making parathesizing a formula more complicated.

Clarification: a formula should be paranthesized such that the scope of a bound variable extends as far to the right as possible.

For example, the formula:

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

should be paranthesized as follows:

$$\Big(\forall x.\big(P(x,x) \vee (\neg(\exists y.(P(x,y) \wedge P(x,x))))\big)\Big).$$

3.13 Exercises

Exercise 32. *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 33. *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* Σ *so that the structure* S *is a* Σ *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 34. *Give examples of* 5 *terms over the signature in Exercise 33 (1.) and compute their abstract syntax tree.*

Exercise 35. *Give examples of* 5 *formulas over the signature in ExerciseExercise 33 (1.) and compute their abstract syntax tree.*

Exercise 36. *Compute the abstract syntax tree of the following formulae (hint: place brackets around subformulae, in the priority order of the logical connectives):*

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. $\forall \mathtt{x}'.\neg\forall \mathtt{x}.\mathtt{P}(\mathtt{x}) \wedge \exists \mathtt{y}.\mathtt{Q}(\mathtt{x},\mathtt{y}) \vee \neg \mathtt{Q}(\mathtt{z},\mathtt{z}) \rightarrow \exists \mathtt{z}'.\mathtt{P}(\mathtt{z}')$.

Exercise 37. *Mark the free occurrences and respectively the bound occurrences of the variables in the formulae below:*

1. $\varphi_1 = (\forall \mathtt{x}.\mathtt{P}(\mathtt{x},\mathtt{x}) \wedge \mathtt{P}(\mathtt{x},\mathtt{y})) \wedge \mathtt{P}(\mathtt{x},\mathtt{z})$;
2. $\varphi_2 = (\forall \mathtt{x}.\mathtt{P}(\mathtt{f}(\mathtt{x},\mathtt{x}),\mathtt{i}(\mathtt{x})) \wedge \exists \mathtt{y}.(\mathtt{P}(\mathtt{x},\mathtt{y}) \wedge \mathtt{P}(\mathtt{x},\mathtt{z})))$.

Exercise 38. *Identify the scope of the quantifiers in the formulae φ_1 and φ_2 from Exercise 37.*

Exercise 39. *Compute the variables, the free variables and respectively the bound variables in the formulae φ_1 and φ_2 from Exercise 37.*

Exercise 40. *Let $A = \{\mathtt{p},\mathtt{q},\mathtt{r},\ldots\}$ be the set of propositional variables. We consider the signature $\Sigma_{\mathbb{PL}} = (A,\emptyset)$, where the propositional variables in A are predicate symbols of arity* 0.

1. *Prove that for any formula $\varphi \in \mathbb{PL}$ we have $\varphi \in \mathbb{FOL}$ (over signature $\Sigma_{\mathbb{PL}}$).*
2. *Prove that for any cuantifier free formula $\varphi \in \mathbb{FOL}$ over $\Sigma_{\mathbb{PL}}$, we have $\varphi \in \mathbb{PL}$.*

Chapter 4

The Semantics of First-Order Formulae

The syntax of first-order logic tells us what are, from a syntactic point of view, the formulae of first-order logic. On the other hand, the semantics of the logic refers to the *meaning* of formulas. The semantics of a formula (or its meaning) shall be 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 the *mathematical structure* over 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 associated natural number called its 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 functional 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 Σ-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 41. *We show several examples of Σ-structures below:*

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 associated to the functional symbol* f*, $-$ is the unary minus function associated to the functional 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 functional symbol* f*, $\cdot^{-1}$ is the unary function associated to the functional 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 functional 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 functional 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 functional symbol* f*, s is the successor function associated to the functional 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 42 (Assignment)**.** *Let Σ be a signature and let S be Σ-structure with the domain D.*

An S-assignment is any function

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

Example 43. *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 44. *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 α we can compute the value of a term in the assignment α. For this, we use an extension of α, denoted $\overline{\alpha}$,

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

which is defined below.

Definition 45 (The Value of a Term in an Assignment)**.** *Given an S-assignment α and a term $t \in \mathcal{T}$ over the signature Σ,* the value of the term t in the assignment α *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 46. *Continuing Example 43, where α_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 α_1 is -5.

Definition 47 (Updating an Assignment)**.** *Given an assignment α, a variable $x \in \mathcal{X}$ and a value $u \in D$, we denote by $\alpha[x \mapsto u]$ a new assignment, obtained from α 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 48. *For example, the assignment $\alpha_1[\mathsf{x}_1 \mapsto 6]$ is exactly the same as the assignment α_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 49. *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 Σ be a signature, let S be a Σ-structure and let α be an S-assignment. Having fixed the elements above, we may compute the truth value of a first-order formula constructed over the signature Σ.

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

that the formula φ is not true (is false) in the structure S with the assignment α.

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

Definition 50. *The fact that a structure S satisfies a formula φ with an assignment α (or equivalently, that φ is true in the structure S with the assignment α) 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* $u \in D$, *such that* $S, \alpha[x \mapsto u] \models \varphi$.

Example 51. *We shall work over the signature* $\Sigma = (\{\mathtt{P}\}, \{\mathtt{f}, \mathtt{i}, \mathtt{e}\})$, *the* Σ*-structure* $S_1 = (\mathbb{Z}, \{=\}, \{+, -, 0\})$ *and the* S_1*-assignments* α_1, α_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* α_1. *Equivalently, we say that* S_1 *satisfies* $\mathtt{P}(\mathtt{x}_1, \mathtt{x}_1)$ *with the assignment* α_1.

Example 52. *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* α_1. *Equivalently, we say that* S_1 *does not satisfy* $\mathtt{P}(\mathtt{x}_1, \mathtt{x}_3)$ *with the assignment* α_1.

Example 53. *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* α_1. *Equivalently, we say that* S_1 *satisfies* $\neg \mathtt{P}(\mathtt{x}_1, \mathtt{x}_3)$ *with the assignment* α_1.

Example 54. *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 55. *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 56. *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 57. *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 58. *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 59 (Satisfiability in a Fixed Structure)**.** *A formula φ* is satisfiable in a structure *S if there exists an S-assignment α such that*

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

Example 60. *The formula $\mathtt{P}(x_1, x_3)$ is satisfiable in the structure S_1 because there is an assignment, for instance α_2, such that $S_1, \alpha_2 \models \mathtt{P}(x_1, x_3)$.*

Exercise 61. *Show that $\neg\mathtt{P}(x_1, x_1)$ is not satisfiable in S_1 (because, for each assignment α, we have $S_1, \alpha \not\models \neg\mathtt{P}(x_1, x_1)$).*

4.4 Validity in a Fixed Structure

Definition 62 (Validity in a Fixed Structure)**.** *A formula φ* is valid in a structure *S if, for any S-assignment α, we have that*

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

Example 63. *The formula $\mathtt{P}(x_1, x_3)$ is not valid in the structure S_1, because there exists an assignment, namely α_1, having the property that $S_1, \alpha_1 \not\models \mathtt{P}(x_1, x_3)$.*

Exercise 64. *Show that $\mathtt{P}(x_1, x_1)$ is valid in the structure S_1 (because, for any assignment α, $S_1, \alpha \models \mathtt{P}(x_1, x_1)$).*

4.5 Satisfiability

Definition 65 (Satisfiability)**.** *A formula φ* is satisfiable *if there exists a structure S and an S-assignment α such that*

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

Example 66. *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 α_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 α_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 67 (Validity)**.** *A formula φ* is valid *if, for any structure S and for any S-assignment α, we have that*

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

Example 68. *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 69. *A formula φ* 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 α for which $S, \alpha \models \varphi_1$, $S, \alpha \models \varphi_2$, $\ldots$, $S, \alpha \models \varphi_n$, we also have that $S, \alpha \models \varphi$.*

Example 70. *We have $\mathtt{P}(\mathtt{x}, \mathtt{y}) \models_{S_1} \mathtt{P}(\mathtt{y}, \mathtt{x})$, because, for any S_1-assignment α 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 71. *A formula φ 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 72. *We have $\mathtt{P(x,y)} \not\models \mathtt{P(y,x)}$, because there exists a structure (namely S_5) so that $\mathtt{P(x,y)} \not\models_{S_5} \mathtt{P(y,x)}$.*

Exercise 73. *Show that*

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

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 Exercises

We recall here the structures show in Example 41:

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 74. *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 75. *Find for each item below a S_2-assignment α_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 76. *Show that the formulas 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 77. *Show that $\forall \mathtt{x}.\exists \mathtt{y}.\mathtt{P}(\mathtt{x}, \mathtt{i}(\mathtt{y}))$ is not valid in S_3.*

Exercise 78. *Find a formula that is satisfiable in S_1 but not satisfiable in S_3.*

Exercise 79. *Find a formula wihout free variables which is satisfiable in S_5 but it is not satisfiable in S_4.*

Exercise 80. *Show that $\forall \mathtt{x}.\exists \mathtt{y}.\mathtt{P}(\mathtt{x}, \mathtt{y})$ is not valid.*

Exercise 81. *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 82. *Show that $\forall \mathtt{x}.\exists \mathtt{y}.\mathtt{P}(\mathtt{y}, \mathtt{x})$ is not valid.*

Exercise 83. *Show that $\forall \mathtt{x}.\neg \mathtt{P}(\mathtt{x}, \mathtt{x})$ is satisfiable.*

Exercise 84. *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.*

Exercise 85. *In Exercise 40 (in Chapter 3) we have shown that $\mathbb{FOL}$ is a syntactic extension of $\mathbb{PL}$. Briefly, if $A = \{\mathtt{p}, \mathtt{q}, \mathtt{r}, \ldots\}$ is a set of propositional variables then we build a signature $\Sigma_{\mathbb{PL}} = \{A, \emptyset\}$, where the propositional variables in A are predicate symbols of arity 0.*

The semantics of $\mathbb{FOL}$ formulas built over the signature $\Sigma_{\mathbb{PL}}$ is consistent with the semantics of $\mathbb{PL}$ formulas. Let $\tau : A \rightarrow B$ be an assigment. Let $S = (D, \{a^S \mid a \in A\}, \emptyset)$ a $\Sigma_{\mathtt{LP}}$-structure, where D is any nonempty set and $a^S = \tau(a)$, for all $a \in A$.

Prove that for any $\varphi \in \mathbb{PL}$, we have $\tau \models \varphi$ iff $S, \alpha \models \varphi$ for all S-assignments $\alpha : \mathcal{X} \rightarrow D$.

Chapter 5

Natural Deduction

In this chapter we present the natural deduction for first-order logic. We define the notion of *substitution*, we recall several notions that we discussed for natural deduction in propositional logic and we present an extended deductive system together with their soundness and completeness properties.

Remark. *The rules of the deductive system of the natural deduction includes the rules presented for propositional logic. The latter are explained again and exemplified on first-order formulas. The natural deduction of* $\mathbb{FOL}$ *also includes rules for cuantifiers. These rules are completely new.*

5.1 Substitutions

Definition 86. *A* substitution *is a function* $\sigma : \mathcal{X} \to \mathcal{T}$*, with the property that* $\sigma(x) \neq x$ *for a finite number of variables* $x \in \mathcal{X}$*.*

Definition 87. *If* $\sigma : \mathcal{X} \to \mathcal{T}$ *is a substitution, the set* $dom(\sigma) = \{x \in \mathcal{X} \mid \sigma(x) \neq x\}$ *is the* domain of the substitution σ*.*

Remark. *By definition, the domain of a substitution is a finite set.*

Definition 88. *If* $\sigma : \mathcal{X} \to \mathcal{T}$ *is a substitution, then the unique* extension *of the substitution* σ *to the set of terms is the function* $\sigma^\sharp : \mathcal{T} \to \mathcal{T}$*, recursively defined as follows:*

1. $\sigma^\sharp(x) = \sigma(x)$*, for any* $x \in \mathcal{X}$*;*
2. $\sigma^\sharp(c) = c$*, for any constant symbol* $c \in \mathcal{F}_0$*;*
3. $\sigma^\sharp(f(t_1, \ldots, t_n)) = f(\sigma^\sharp(t_1), \ldots, \sigma^\sharp(t_n))$*, for any functional symbol* $f \in \mathcal{F}_n$ *of arity* $n \in \mathbb{N}^*$ *and any terms* $t_1, \ldots, t_n \in \mathcal{T}$*.*

The substitutions are noted by $\sigma, \tau, \sigma_0, \tau_1, \sigma'$, etc.

Remark. *If $t \in \mathcal{T}$ is a term, then $\sigma^\sharp(t) \in \mathcal{T}$ is the term obtained from t by* applying *to the substitution σ or the term obtained by* applying the substitution σ on the term t.

Practically, in order to obtain $\sigma^\sharp(t)$ from t, all occurences of a variable x in t are replaced simultaneously with the corresponding term $\sigma(x)$.

Example 89. *Let consider the substitution $\sigma_1 : \mathcal{X} \to \mathcal{T}$ defined as follows:*

1. $\sigma_1(\mathsf{x}_1) = \mathsf{x}_2$;
2. $\sigma_1(\mathsf{x}_2) = \mathsf{f}(\mathsf{x}_3, \mathsf{x}_4)$;
3. $\sigma_1(x) = x$ *for all* $x \in \mathcal{X} \setminus \{\mathsf{x}_1, \mathsf{x}_2\}$.

Let consider the term $t = \mathsf{f}(\mathsf{f}(\mathsf{x}_1, \mathsf{x}_2), \mathsf{f}(\mathsf{x}_3, \mathsf{e}))$. We have that:

$$\begin{aligned}
\sigma_1^\sharp\Big(t\Big) &= \sigma_1^\sharp\Big(\mathsf{f}(\mathsf{f}(\mathsf{x}_1, \mathsf{x}_2), \mathsf{f}(\mathsf{x}_3, \mathsf{e}))\Big) \\
&= \mathsf{f}(\sigma_1^\sharp\Big(\mathsf{f}(\mathsf{x}_1, \mathsf{x}_2)\Big), \sigma_1^\sharp\Big(\mathsf{f}(\mathsf{x}_3, \mathsf{e})\Big)) \\
&= \mathsf{f}(\mathsf{f}(\sigma_1^\sharp\Big(\mathsf{x}_1\Big), \sigma_1^\sharp\Big(\mathsf{x}_2\Big)), \mathsf{f}(\sigma_1^\sharp\Big(\mathsf{x}_3\Big), \sigma_1^\sharp\Big(\mathsf{e}\Big))) \\
&= \mathsf{f}(\mathsf{f}(\sigma_1\Big(\mathsf{x}_1\Big), \sigma_1\Big(\mathsf{x}_2\Big)), \mathsf{f}(\sigma_1\Big(\mathsf{x}_3\Big), \mathsf{e})) \\
&= \mathsf{f}(\mathsf{f}(\mathsf{x}_2, \mathsf{f}(\mathsf{x}_3, \mathsf{x}_4)), \mathsf{f}(\mathsf{x}_3, \mathsf{e})).
\end{aligned}$$

Note that by applying a substitution over a term, we replace (in the same time) all occurrences of the variables from the domain of the substitution with the corresponding terms.

Notation. *If $dom(\sigma) = \{x_1, \ldots, x_n\}$, then the substitution σ cam be also written as:*

$$\sigma = \{x_1 \mapsto \sigma(x_1), \ldots, x_n \mapsto \sigma(x_n)\}.$$

Attention, it is not a set, but only some notation for substitutions.

Example 90. *For the substitution from the previous example, we have*

$$\sigma_1 = \{\mathsf{x}_1 \mapsto \mathsf{x}_2, \mathsf{x}_2 \mapsto \mathsf{f}(\mathsf{x}_3, \mathsf{x}_4)\}.$$

Definition 91. *If $\sigma : \mathcal{X} \to \mathcal{T}$ is a substitution and $V \subseteq \mathcal{X}$ is a subset of variables, then* the restriction of the substitution σ to the set V is another substitution $\sigma|_V : \mathcal{X} \to \mathcal{T}$, *defined as follows:*

1. $\sigma|_V(x) = \sigma(x)$ *for any* $x \in V$;

2. $\sigma|_V(x) = x$ *for any* $x \in \mathcal{X} \setminus V$.

Example 92. *For the substitution* σ_1 *in Example 89, we have* $\sigma_1|_{\{x_1\}} = \{x_1 \mapsto x_2\}$ *și* $\sigma_1|_{\{x_2\}} = \{x_2 \mapsto f(x_3, x_4)\}$.

In other words, by restricting a substitution to a set of variables, we remove some the other variables from the domain of the substitution.

Definition 93. *for any substitution* $\sigma : \mathcal{X} \to \mathcal{T}$, the extention *of* σ *to the set of formulae is the function* $\sigma^\flat : \mathbb{FOL} \to \mathbb{FOL}$, *defined as:*

1. $\sigma^\flat(P(t_1, \ldots, t_n)) = P(\sigma^\sharp(t_1), \ldots, \sigma^\sharp(t_n))$;
2. $\sigma^\flat(\neg\varphi) = \neg\sigma^\flat(\varphi)$;
3. $\sigma^\flat\big((\varphi_1 \wedge \varphi_2)\big) = (\sigma^\flat(\varphi_1) \wedge \sigma^\flat(\varphi_2))$;
4. $\sigma^\flat\big((\varphi_1 \vee \varphi_2)\big) = (\sigma^\flat(\varphi_1) \vee \sigma^\flat(\varphi_2))$;
5. $\sigma^\flat\big((\varphi_1 \to \varphi_2)\big) = (\sigma^\flat(\varphi_1) \to \sigma^\flat(\varphi_2))$;
6. $\sigma^\flat\big((\varphi_1 \leftrightarrow \varphi_2)\big) = (\sigma^\flat(\varphi_1) \leftrightarrow \sigma^\flat(\varphi_2))$;
7. $\sigma^\flat\big((\forall x.\varphi)\big) = \big(\forall x.(\rho^\flat(\varphi))\big)$, *where* $\rho = \sigma|_{dom(\sigma)\setminus\{x\}}$;
8. $\sigma^\flat\big((\exists x.\varphi)\big) = \big(\exists x.(\rho^\flat(\varphi))\big)$, *where* $\rho = \sigma|_{dom(\sigma)\setminus\{x\}}$;

In other words, to obtain the formula $\sigma^\flat(\varphi)$ from φ, every free occurrence of the variable x from the formula φ is replaced by the term $\sigma(x)$.

Example 94. *Using the substitution* σ_1 *in Example 89, we have that:*

$$\begin{aligned}
\sigma_1^\flat\Big((\forall x_2.P(x_1, x_2)) \wedge P(x_2, x_2)\Big) &= \\
\sigma_1^\flat\Big((\forall x_2.P(x_1, x_2))\Big) \wedge \sigma_1^\flat\Big(P(x_2, x_2)\Big) &= \\
(\forall x_2.\sigma_1|^\flat_{\{x_1\}}\Big(P(x_1, x_2)\Big)) \wedge P(\sigma_1^\sharp\Big(x_2\Big), \sigma_1^\sharp\Big(x_2\Big)) &= \\
(\forall x_2.P(\sigma_1|^\sharp_{\{x_1\}}\Big(x_1\Big), \sigma_1|^\sharp_{\{x_1\}}\Big(x_2\Big))) \wedge P(\sigma_1\Big(x_2\Big), \sigma_1\Big(x_2\Big)) &= \\
(\forall x_2.P(\sigma_1|_{\{x_1\}}\Big(x_1\Big), \sigma_1|_{\{x_1\}}\Big(x_2\Big))) \wedge P(f(x_3, x_4), f(x_3, x_4)) &= \\
(\forall x_2.P(\sigma_1\Big(x_1\Big), x_2)) \wedge P(f(x_3, x_4), f(x_3, x_4)) &= \\
(\forall x_2.P(x_2, x_2)) \wedge P(f(x_3, x_4), f(x_3, x_4)). &
\end{aligned}$$

Remark. *Attention: the bound occurrences of variables are NOT replaced when applying the substitution! In Example 94, the occurrence of the variable* x_2 *in* $(\forall x_2.P(x_1, x_2))$ *is bound.*

Notation. *According to Notation 5.1, for the substitutions with a finite domain, we also use the notation* $\{x_1 \mapsto \sigma(x_1), \ldots, x_n \mapsto \sigma(x_n)\}$. *We will also use substitutions without associating a name to them, since they are simple having the form :* $\{x \mapsto t\}$. *In order to express the fact that we apply this substitution to a formula, according to our notation, we should write* $\{x \mapsto t\}(\varphi)$. *However, in the literature other notations are preferred that we will also use. One variant is to write* $\varphi[t/x]$. *Another one is* $\varphi[x \mapsto t]$. *In this document we will use the latter notation, namely* $\varphi[x \mapsto t]$.

5.2 Sequences

Definition 95 (Sequence)**.** *A* sequence *is a pair formed by a set of formulae* $\{\varphi_1, \ldots, \varphi_n\} \subseteq \mathbb{FOL}$ *and a formula* $\varphi \in \mathbb{FOL}$, *denoted by:*

$$\{\varphi_1, \ldots, \varphi_n\} \vdash \varphi.$$

sometimes we read the notation $\{\varphi_1, \ldots, \varphi_n\} \vdash \varphi$ as φ *is a syntactic consequence from* $\{\varphi_1, \ldots, \varphi_n\}$. Usually, we will note with $\Gamma = \{\varphi_1, \ldots, \varphi_n\}$ the set of hypothesis and we will write the sequence as $\Gamma \vdash \varphi$.

Remark. *We recall that the usual notation in the literature allows us to write* $\varphi_1, \ldots, \varphi_n \vdash \varphi$ *(without curly brackets) instead of* $\{\varphi_1, \ldots, \varphi_n\} \vdash \varphi$. *However, we have to remember that on the left side of the symbol* $\vdash$ *is all the time a set. The notation without brackets allows us to write* $\varphi_1, \ldots, \varphi_n, \psi \vdash \varphi$ *instead of* $\{\varphi_1, \ldots, \varphi_n\} \cup \{\psi\} \vdash \varphi$.

Example 96. *In many examples from this material, we work with the signature* $\Sigma = (\{\mathtt{P}, \mathtt{Q}\}, \{\mathtt{a}, \mathtt{b}, \mathtt{f}, \mathtt{g}\})$, *where the predicate symbols* $\mathtt{P}$ *and* $\mathtt{Q}$ *have arity 1, the functional symbols* $\mathtt{f}$ *and* $\mathtt{g}$ *have arity 1, and the symbols* $\mathtt{a}$ *and* $\mathtt{b}$ *are constants (of arity 0).*

Example 97. *Let consider the signature* Σ *from Example 96. The following are some examples of sequences:*

1. $\{\mathtt{P}(\mathtt{a}), \mathtt{Q}(\mathtt{a})\} \vdash (\mathtt{P}(\mathtt{a}) \wedge \mathtt{Q}(\mathtt{a}))$;
2. $\{\forall \mathtt{x}.\mathtt{Q}(\mathtt{x}), \mathtt{P}(\mathtt{a})\} \vdash (\mathtt{P}(\mathtt{a}) \wedge \mathtt{Q}(\mathtt{a}))$;
3. $\{\exists \mathtt{x}.\mathtt{Q}(\mathtt{x})\} \vdash \mathtt{Q}(\mathtt{a})$.

Later we will see that the first two sequences from above are valid,*and the last one is not valid.*

It is often convenient to write sequence without curly braces as in the next example:

Example 98. *The sequences in Example 97 can be written without curly braces as follows:*

1. $\mathtt{P(a)}, \mathtt{Q(a)} \vdash (\mathtt{P(a)} \wedge \mathtt{Q(a)})$;
2. $\forall \mathtt{x}.\mathtt{Q(x)}, \mathtt{P(a)} \vdash (\mathtt{P(a)} \wedge \mathtt{Q(a)})$;
3. $\exists \mathtt{x}.\mathtt{Q(x)} \vdash \mathtt{Q(a)}$.

5.3 Inference rules

Definition 99. *An* inference rule *is a tuple formed from:*

1. *a set of sequences* $S_1, \ldots, S_n$, *called* hypothesis *of the rule;*
2. *a sequence* S *called* conclusion *of the rule;*
3. *a* condition *for the applicability of the rule;*
4. *a name.*

An inference rule is noted as follows:

$$\textsc{name}\ \frac{S_1 \qquad \ldots \qquad S_n}{S}\ \textit{condition.}$$

Remark. *The inference rules that have* $n = 0$ *hypothesis, are called axioms. Also, the applicability conditions may be absent.*

Example 100. *The following are some inference rules from propositional logic:*

$$\wedge i\ \frac{\Gamma \vdash \varphi_1 \qquad \Gamma \vdash \varphi_2}{\Gamma \vdash (\varphi_1 \wedge \varphi_2),} \qquad \wedge e_1\ \frac{\Gamma \vdash (\varphi_1 \wedge \varphi_2)}{\Gamma \vdash \varphi_1,} \qquad \wedge e_2\ \frac{\Gamma \vdash (\varphi_1 \wedge \varphi_2)}{\Gamma \vdash \varphi_2.}$$

As in the case of propositional logic, all three inference rules from above are correct. *None of them has a applicability condition. The following is an example of inference rule with* $n = 0$ *hypothesis, but with one condition.*

$$\textsc{Hypothesis}\ \frac{}{\Gamma \vdash \varphi}\ \varphi \in \Gamma.$$

Below we have an example of incorrect inference rule (in a way that we will clarify later, but that can be already perceived).

$$\text{REGULĂ INCORECTĂ}\ \frac{\Gamma \vdash \varphi_2}{\Gamma \vdash (\varphi_1 \wedge \varphi_2).}$$

Remark. *The hypothesis of the inference rule, as the conclusion, are in fact* patterns *for sequences and not the sequences themselves. These patterns can be instantiated, meaning that an inference rule (presented above) has several* instances *obtained by replacing mathematical variables $\varphi, \varphi', \Gamma$ with concrete formulae. For instance, here is a new instance for the rule $\wedge i$ from above:*

$$\wedge i\ \frac{\{\mathtt{P(a)}, \mathtt{Q(a)}\} \vdash \mathtt{P(a)} \qquad \{\mathtt{P(a)}, \mathtt{Q(a)}\} \vdash \mathtt{Q(a)}}{\{\mathtt{P(a)}, \mathtt{Q(a)}\} \vdash (\mathtt{P(a)} \wedge \mathtt{Q(a)});}$$

$$\wedge i\ \frac{\{\mathtt{P(a)}, \mathtt{Q(a)}, \mathtt{Q(b)}\} \vdash (\mathtt{P(a)} \wedge \mathtt{Q(a)}) \qquad \{\mathtt{P(a)}, \mathtt{Q(a)}, \mathtt{Q(b)}\} \vdash \mathtt{P(a)}}{\{\mathtt{P(a)}, \mathtt{Q(a)}, \mathtt{Q(b)}\} \vdash ((\mathtt{P(a)} \wedge \mathtt{Q(a)}) \wedge \mathtt{P(a)}).}$$

We first replaced the mathematical variable Γ with the set of formulae $\{\mathtt{P(a)}, \mathtt{Q(a)}\}$, the mathematical variable φ with the formula $\mathtt{P(a)}$ and the mathematical variable φ' with the formula $\mathtt{Q(a)}$. Exercise: establish with what was replaced each mathematical variable from the second instance.

Here is an example of rule that is not an instance of a rule $\wedge i$ (exercise: explain why not):

$$?\ \frac{\{\mathtt{P(a)}, \mathtt{Q(a)}\} \vdash \mathtt{P(a)} \qquad \{\mathtt{P(a)}, \mathtt{Q(a)}\} \vdash \mathtt{Q(a)}}{\{\mathtt{P(a)}, \mathtt{Q(a)}\} \vdash (\mathtt{P(a)} \wedge \mathtt{Q(a)});}$$

5.4 Deductive system

Definition 101. *A* deductive system *is a set of inference rules.*

Example 102. *Let consider the deductive system D_1, formed from the following four inference rules:*

$$\text{HYPOTHESIS}\ \frac{}{\Gamma \vdash \varphi,}\ \varphi \in \Gamma \qquad \wedge i\ \frac{\Gamma \vdash \varphi_1 \qquad \Gamma \vdash \varphi_2}{\Gamma \vdash (\varphi_1 \wedge \varphi_2),} \qquad \wedge e_1\ \frac{\Gamma \vdash (\varphi_1 \wedge \varphi_2)}{\Gamma \vdash \varphi_1,}$$

$$\wedge e_2\ \frac{\Gamma \vdash (\varphi_1 \wedge \varphi_2)}{\Gamma \vdash \varphi_2.}$$

5.5 Formal proof

Definition 103 (Formal proof)**.** *A* formal proof *in a deductive system is a list of sequences*

1. S_1
2. S_2

...

n. S_n,

with the property that each sequence S_i is justified by a inference rule of the deductive system applied on the previous sequences $(S_1, \ldots, S_{i-1})$, meaning that S_i is the conclusion of an instance of an inference rule from the deductive rule, rule that uses as hypothesis sequences chosen from $S_1, \ldots, S_{i-1}$. In addition, if the inference rule has a condition, this condition has to be true. Note also that any prefix of a formal proof is also a proof.

Example 104. *Here is an example of formal proof in the deductive system D_1 from above:*

1. $\{\mathtt{P(a)}, \mathtt{Q(a)}\} \vdash \mathtt{P(a)}$; *(Hypothesis)*
2. $\{\mathtt{P(a)}, \mathtt{Q(a)}\} \vdash \mathtt{Q(a)}$; *(Hypothesis)*
3. $\{\mathtt{P(a)}, \mathtt{Q(a)}\} \vdash (\mathtt{P(a)} \wedge \mathtt{Q(a)})$; *($\wedge i$, 1, 2)*
4. $\{\mathtt{P(a)}, \mathtt{Q(a)}\} \vdash (\mathtt{Q(a)} \wedge (\mathtt{P(a)} \wedge \mathtt{Q(a)}))$. *($\wedge i$, 2, 3)*

As in the case of propositional logic, each line has the name of the applied inference rule and the lines where the needed hypothesis are found (in the same order used to present the deductive system).

Remark. *The definition of the formal proof in the first order logic is the same as in the case of propositional logic. However, we will see later that in order to apply the new inference rules, associated to the quantifiers, we will use new annotations for the lines of the formal proof.*

Definition 105 (Valid sequence)**.** *A sequence $\Gamma \vdash \varphi$ is* valid in a deductive system D *if there is a formal proof $S_1, \ldots, S_n$ in D such that $S_n = \Gamma \vdash \varphi$.*

Example 106. *The sequence $\{\mathtt{P(a)}, \mathtt{Q(a)}\} \vdash (\mathtt{P(a)} \wedge \mathtt{Q(a)})$ is valid in the deductive system D_1 from above because is the last sequence from the following formal proof:*

1. $\{\mathtt{P(a)}, \mathtt{Q(a)}\} \vdash \mathtt{P(a)}$; *(Hypothesis)*

2. $\{\mathtt{P(a)}, \mathtt{Q(a)}\} \vdash \mathtt{Q(a)}$; *(Hypothesis)*

3. $\{\mathtt{P(a)}, \mathtt{Q(a)}\} \vdash (\mathtt{P(a)} \wedge \mathtt{Q(a)})$; *($\wedge i$, 1, 2)*

Remark. *Attention! Do not mix the notions of* valid sequence in a deductive system *and the notion of* valid formula.

5.6 Natural deduction

Natural deduction is a deductive system for the first order logic. In other words, the deductive system for first order logic includes all the rules of natural deduction from propositional logic. In addition, for first order logic we have new rules for the introduction and elimination of quantifiers. In this section we will present each inference rule from natural deduction of first order logic.

5.6.1 Rules for conjunctions

We already saw the inference rules for the introduction and elimination for the "and" connector:

$$\wedge i \ \frac{\Gamma \vdash \varphi_1 \qquad \Gamma \vdash \varphi_2}{\Gamma \vdash (\varphi_1 \wedge \varphi_2),} \qquad \wedge e_1 \ \frac{\Gamma \vdash (\varphi_1 \wedge \varphi_2)}{\Gamma \vdash \varphi_1,} \qquad \wedge e_2 \ \frac{\Gamma \vdash (\varphi_1 \wedge \varphi_2)}{\Gamma \vdash \varphi_2.}$$

This deductive system is called *natural* because the inference rules mimic the reasoning of humans, based in essence on an intuitive semantics for the notion of truth:

1. The rule for the introduction of the connector $\wedge$ indicates that we can prove a conjunction $(\varphi_1 \wedge \varphi_2)$ from the set of hypothesis Γ if we already know that each part of the conjunction, φ_1 and respectively φ_2, are consequences of the hypotheses from Γ.

   In other words, in order to prove a conjunction from a set of hypotheses, is enough to establish individually that each part of the conjunction is a consequence of the hypothesis.

2. For the $\wedge$ connector, we have two rules for elimination. First elimination rule for the $\wedge$ connector says that if we already established that some conjunction $(\varphi_1 \wedge \varphi_2)$ is the consequence of a set Γ of hypotheses, then the left side of the conjunction, φ_1, is a consequence of the set Γ.

   The second rule is symmetric with respect to the first and says that we can conclude that the right side of the conjunction is the consequence of a set of formulae if the conjunction is is the consequence of this set of formulae.

Here is an example of formal proof that uses the inference rules for the connector $\wedge$:

1. $\{(\mathtt{P}(\mathtt{a}) \wedge \mathtt{Q}(\mathtt{a})), \forall \mathtt{x}.\mathtt{P}(\mathtt{x})\} \vdash (\mathtt{P}(\mathtt{a}) \wedge \mathtt{Q}(\mathtt{a}))$; (HYPOTHESIS)
2. $\{(\mathtt{P}(\mathtt{a}) \wedge \mathtt{Q}(\mathtt{a})), \forall \mathtt{x}.\mathtt{P}(\mathtt{x})\} \vdash \forall \mathtt{x}.\mathtt{P}(\mathtt{x})$; (HYPOTHESIS)
3. $\{(\mathtt{P}(\mathtt{a}) \wedge \mathtt{Q}(\mathtt{a})), \forall \mathtt{x}.\mathtt{P}(\mathtt{x})\} \vdash \mathtt{P}(\mathtt{a})$; $(\wedge e_1,\ 1)$
4. $\{(\mathtt{P}(\mathtt{a}) \wedge \mathtt{Q}(\mathtt{a})), \forall \mathtt{x}.\mathtt{P}(\mathtt{x})\} \vdash (\mathtt{P}(\mathtt{a}) \wedge \forall \mathtt{x}.\mathtt{P}(\mathtt{x}))$. $(\wedge i,\ 3,\ 2)$

5.6.2 Rules for implication

The rule for the elimination of the implication, also called *modus ponens* in latin, is one of the most important rules of inference that we apply.

$$\rightarrow e \ \frac{\Gamma \vdash (\varphi_1 \rightarrow \varphi_2) \qquad \Gamma \vdash \varphi_1}{\Gamma \vdash \varphi_2}$$

The rule shows that, supposing that we proved $\varphi \rightarrow \varphi'$ (from Γ) and in addition we proved that φ (also from Γ), then we can prove φ' (from Γ).

Here is an example of formal proof that uses the rule for the elimination of the implication:

1. $\{(\mathtt{P}(\mathtt{a}) \rightarrow \forall \mathtt{x}.\mathtt{P}(\mathtt{x})), (\mathtt{P}(\mathtt{a}) \wedge \mathtt{Q}(\mathtt{a}))\} \vdash (\mathtt{P}(\mathtt{a}) \wedge \mathtt{Q}(\mathtt{a}))$; (HYPOTHESIS)
2. $\{(\mathtt{P}(\mathtt{a}) \rightarrow \forall \mathtt{x}.\mathtt{P}(\mathtt{x})), (\mathtt{P}(\mathtt{a}) \wedge \mathtt{Q}(\mathtt{a}))\} \vdash \mathtt{P}(\mathtt{a})$; $(\wedge e_1,\ 1)$
3. $\{(\mathtt{P}(\mathtt{a}) \rightarrow \forall \mathtt{x}.\mathtt{P}(\mathtt{x})), (\mathtt{P}(\mathtt{a}) \wedge \mathtt{Q}(\mathtt{a}))\} \vdash (\mathtt{P}(\mathtt{a}) \rightarrow \forall \mathtt{x}.\mathtt{P}(\mathtt{x}))$; (HYPOTHESIS)
4. $\{(\mathtt{P}(\mathtt{a}) \rightarrow \forall \mathtt{x}.\mathtt{P}(\mathtt{x})), (\mathtt{P}(\mathtt{a}) \wedge \mathtt{Q}(\mathtt{a}))\} \vdash \forall \mathtt{x}.\mathtt{P}(\mathtt{x})$. $(\rightarrow e,\ 3,\ 1)$

This proof shows that the sequence $\{(\mathtt{P}(\mathtt{a}) \rightarrow \forall \mathtt{x}.\mathtt{P}(\mathtt{x})), (\mathtt{P}(\mathtt{a}) \wedge \mathtt{Q}(\mathtt{a}))\} \vdash \forall \mathtt{x}.\mathtt{P}(\mathtt{x})$ is valid, meaning that the formula $\forall \mathtt{x}.\mathtt{P}(\mathtt{x})$ is a consequence of the set of formulae $\{(\mathtt{P}(\mathtt{a}) \rightarrow \forall \mathtt{x}.\mathtt{P}(\mathtt{x})), (\mathtt{P}(\mathtt{a}) \wedge \mathtt{Q}(\mathtt{a}))\}$. Observe the order in which the lines 3 and 1 appear in the explanation for line 4: they follow the same order, fixed by the inference rule.

Exercise 107. *Prove that the following sequences are valid:*

1. $\{((\mathtt{P}(\mathtt{a}) \wedge \mathtt{Q}(\mathtt{a})) \rightarrow \forall \mathtt{x}.\mathtt{P}(\mathtt{x})), \mathtt{P}(\mathtt{a}), \mathtt{Q}(\mathtt{a})\} \vdash \forall \mathtt{x}.\mathtt{P}(\mathtt{x})$;
2. $\{(\mathtt{P}(\mathtt{a}) \rightarrow \forall \mathtt{x}.\mathtt{P}(\mathtt{x})), \mathtt{P}(\mathtt{a}), \mathtt{Q}(\mathtt{a})\} \vdash (\mathtt{Q}(\mathtt{a}) \wedge \forall \mathtt{x}.\mathtt{P}(\mathtt{x}))$.

The rule for introducing the implication is subtle. In order to prove that an implication $(\varphi_1 \to \varphi_2)$ follows from Γ, *we suppose* φ_1 (besides Γ) and prove φ_2. In other words, in the hypothesis for the rule, we add the formula φ_1 to the formulae from Γ. The rule may be written in two equivalent ways, that differ only by the fact that the first rule uses the convention referring to the curly brackets around premises from the notations of sequences, while in the second one the brackets appear explicitly:

$$\to i \ \frac{\Gamma, \varphi_1 \vdash \varphi_2}{\Gamma \vdash (\varphi_1 \to \varphi_2),} \qquad\qquad \to i \ \frac{\Gamma \cup \{\varphi_1\} \vdash \varphi_2}{\Gamma \vdash (\varphi_1 \to \varphi_2).}$$

It is important to observe and understand for rule of introduction of implication is that the set of premises changes. In the conclusion we have that the formula $(\varphi_1 \to \varphi_2)$ follows from Γ, while in the hypothesis we have to prove that φ_2 follows from the premises $\Gamma \cup \{\varphi_1\}$. In other words, intuitively, in order to prove an implication $(\varphi_1 \to \varphi_2)$, we suppose the antecedent φ_1 and prove φ_2.

Example 108. *Let's prove that the sequence* $\{\} \vdash (\mathtt{P(a)} \to \mathtt{P(a)})$ *is valid:*

1. $\{\mathtt{P(a)}\} \vdash \mathtt{P(a)}$; *(Hypothesis)*
2. $\{\} \vdash (\mathtt{P(a)} \to \mathtt{P(a)})$. *$(\to i,\ 1)$*

Example 109. $\{(\mathtt{P(a)} \to \mathtt{Q(a)}), (\mathtt{Q(a)} \to \mathtt{P(b)})\} \vdash (\mathtt{P(a)} \to \mathtt{P(b)})$ *is valid:*

1. $\{(\mathtt{P(a)} \to \mathtt{Q(a)}), (\mathtt{Q(a)} \to \mathtt{P(b)}), \mathtt{P(a)}\} \vdash (\mathtt{P(a)} \to \mathtt{Q(a)})$; *(Hypothesis)*
2. $\{(\mathtt{P(a)} \to \mathtt{Q(a)}), (\mathtt{Q(a)} \to \mathtt{P(b)}), \mathtt{P(a)}\} \vdash \mathtt{P(a)}$; *(Hypothesis)*
3. $\{(\mathtt{P(a)} \to \mathtt{Q(a)}), (\mathtt{Q(a)} \to \mathtt{P(b)}), \mathtt{P(a)}\} \vdash \mathtt{Q(a)}$; *$(\to e,\ 1,\ 2)$*
4. $\{(\mathtt{P(a)} \to \mathtt{Q(a)}), (\mathtt{Q(a)} \to \mathtt{P(b)}), \mathtt{P(a)}\} \vdash (\mathtt{Q(a)} \to \mathtt{P(b)})$; *(Hypothesis)*
5. $\{(\mathtt{P(a)} \to \mathtt{Q(a)}), (\mathtt{Q(a)} \to \mathtt{P(b)}), \mathtt{P(a)}\} \vdash \mathtt{P(b)}$; *$(\to e,\ 4,\ 3)$*
6. $\{(\mathtt{P(a)} \to \mathtt{Q(a)}), (\mathtt{Q(a)} \to \mathtt{P(b)})\} \vdash (\mathtt{P(a)} \to \mathtt{P(b)})$. *$(\to i,\ 5)$*

Exercise 110. *Prove that the following sequences are valid:*

1. $\{((\mathtt{P(a)} \wedge \mathtt{Q(a)}) \to \mathtt{P(b)}), \mathtt{P(a)}, \mathtt{Q(a)}\} \vdash \mathtt{P(b)}$;
2. $\{((\mathtt{P(a)} \wedge \mathtt{Q(a)}) \to \mathtt{P(b)})\} \vdash (\mathtt{P(a)} \to (\mathtt{Q(a)} \to \mathtt{P(b)}))$;
3. $\{(\mathtt{P(a)} \to (\mathtt{Q(a)} \to \mathtt{P(b)}))\} \vdash ((\mathtt{P(a)} \wedge \mathtt{Q(a)}) \to \mathtt{P(b)})$.

5.6.3 Rules for disjunction

The connector $\vee$ has two introduction rules:

$$\vee i_1 \ \frac{\Gamma \vdash \varphi_1}{\Gamma \vdash (\varphi_1 \vee \varphi_2),} \qquad \vee i_2 \ \frac{\Gamma \vdash \varphi_2}{\Gamma \vdash (\varphi_1 \vee \varphi_2).}$$

The first rule shows that if we know φ_1 (from Γ), then we also know $(\varphi_1 \vee \varphi_2)$ (from Γ), no matter what φ_2 is. The second rule of elimination is symmetric, for the right side of the disjunction.

Example 111. *Let's prove that* $\{(\mathtt{P}(\mathtt{a}) \wedge \mathtt{Q}(\mathtt{a}))\} \vdash (\mathtt{P}(\mathtt{a}) \vee \mathtt{Q}(\mathtt{a}))$ *is valid:*

1. $\{(\mathtt{P}(\mathtt{a}) \wedge \mathtt{Q}(\mathtt{a}))\} \vdash (\mathtt{P}(\mathtt{a}) \wedge \mathtt{Q}(\mathtt{a}))$; *(Hypothesis)*
2. $\{(\mathtt{P}(\mathtt{a}) \wedge \mathtt{Q}(\mathtt{a}))\} \vdash \mathtt{P}(\mathtt{a})$; *($\wedge e_1$, 1)*
3. $\{(\mathtt{P}(\mathtt{a}) \wedge \mathtt{Q}(\mathtt{a}))\} \vdash (\mathtt{P}(\mathtt{a}) \vee \mathtt{Q}(\mathtt{a}))$. *($\vee i_1$, 2)*

Another formal proof of the sequence is the following:

1. $\{(\mathtt{P}(\mathtt{a}) \wedge \mathtt{Q}(\mathtt{a}))\} \vdash (\mathtt{P}(\mathtt{a}) \wedge \mathtt{Q}(\mathtt{a}))$; *(Hypothesis)*
2. $\{(\mathtt{P}(\mathtt{a}) \wedge \mathtt{Q}(\mathtt{a}))\} \vdash \mathtt{Q}(\mathtt{a})$; *($\wedge e_2$, 1)*
3. $\{(\mathtt{P}(\mathtt{a}) \wedge \mathtt{Q}(\mathtt{a}))\} \vdash (\mathtt{P}(\mathtt{a}) \vee \mathtt{Q}(\mathtt{a}))$. *($\vee i_2$, 2)*

The proof for the elimination of disjunction is a little mor complicated, being another rule in which the set of premises of the sequences changes from hypotheses to conclusion:

$$\vee e \ \frac{\Gamma \vdash (\varphi_1 \vee \varphi_2) \qquad \Gamma, \varphi_1 \vdash \varphi' \qquad \Gamma, \varphi_2 \vdash \varphi'}{\Gamma \vdash \varphi'}$$

The first hypothesis of the rule, $\Gamma \vdash (\varphi_1 \vee \varphi_2)$, is easy to understand: in order to "remove" a disjunction, we need a disjunction between the hypotheses (disjunction that we want to "eliminate"). The last two hypothesis of the elimination rule of disjunction has to be understood intuitively as follows. From the first hypothesis we know $(\varphi_1 \vee \varphi_2)$ (from Γ); in other words, at least one of the formulae φ_1 and respectively φ_2 follows from Γ. The hypotheses 2 and 3 indicates that, no matter which of the formulae φ_1 or φ_2 holds, in any case φ' holds. That is, if we suppose φ_1 (besides Γ), φ' holds, and if we suppose φ_2 (besides Γ), φ' still holds. And therefore the conclusion indicates that φ' holds no matter which one of the formulae φ_1 and respectively φ_2 would hold.

Example 112. *Let us prove that the sequence* $\{(\mathtt{P}(\mathtt{a}) \vee \mathtt{Q}(\mathtt{a}))\} \vdash (\mathtt{Q}(\mathtt{a}) \vee \mathtt{P}(\mathtt{a}))$ *is valid:*

1. $\{(\mathtt{P}(\mathtt{a}) \vee \mathtt{Q}(\mathtt{a})), \mathtt{P}(\mathtt{a})\} \vdash \mathtt{P}(\mathtt{a})$; *(Hypothesis)*
2. $\{(\mathtt{P}(\mathtt{a}) \vee \mathtt{Q}(\mathtt{a})), \mathtt{P}(\mathtt{a})\} \vdash (\mathtt{Q}(\mathtt{a}) \vee \mathtt{P}(\mathtt{a}))$; *$(\vee i_2,\ 1)$*
3. $\{(\mathtt{P}(\mathtt{a}) \vee \mathtt{Q}(\mathtt{a})), \mathtt{Q}(\mathtt{a})\} \vdash \mathtt{Q}(\mathtt{a})$; *(Hypothesis)*
4. $\{(\mathtt{P}(\mathtt{a}) \vee \mathtt{Q}(\mathtt{a})), \mathtt{Q}(\mathtt{a})\} \vdash (\mathtt{Q}(\mathtt{a}) \vee \mathtt{P}(\mathtt{a}))$; *$(\vee i_1,\ 1)$*
5. $\{(\mathtt{P}(\mathtt{a}) \vee \mathtt{Q}(\mathtt{a}))\} \vdash (\mathtt{P}(\mathtt{a}) \vee \mathtt{Q}(\mathtt{a}))$; *(Hypothesis)*
6. $\{(\mathtt{P}(\mathtt{a}) \vee \mathtt{Q}(\mathtt{a}))\} \vdash (\mathtt{Q}(\mathtt{a}) \vee \mathtt{P}(\mathtt{a}))$. *$(\vee e,\ 5,\ 2,\ 4)$*

Note the way in which the set of premises changes from a sequence to the other in the formal proof, following the inference rules.

Exercise 113. *Find a formal proof for the sequence*

$$\{(\mathtt{P}(\mathtt{a}) \vee \mathtt{Q}(\mathtt{a})), (\mathtt{P}(\mathtt{a}) \to \mathtt{P}(\mathtt{b})), (\mathtt{Q}(\mathtt{a}) \to \mathtt{P}(\mathtt{b}))\} \vdash \mathtt{P}(\mathtt{b}).$$

5.6.4 Rules for negation

The rules for the introduction end elimination of the negation are presented together with a rule for the elimination of $\bot$:

$$\neg i\ \frac{\Gamma, \varphi \vdash \bot}{\Gamma \vdash \neg\varphi} \qquad \neg e\ \frac{\Gamma \vdash \varphi \qquad \Gamma \vdash \neg\varphi}{\Gamma \vdash \bot} \qquad \bot e\ \frac{\Gamma \vdash \bot}{\Gamma \vdash \varphi}$$

Let us recall that $\bot$ is a logical connector of arity 0. In other words, the connector $\bot$ does not change in the formula. The semantics of the formula $\bot$ is such that it is false in any structure and any assignment. In other words, $\bot$ is a contradiction.

The first rule , for the introduction of negation, is easy to explayn intuitively: how can we prove that a formula of the form $\neg\varphi$ follows from Γ? We suppose, in addition besides Γ, that we have φ and prove that from Γ and φ follows a contradiction ($\Gamma, \varphi \vdash \bot$). In this way, we prove that $\neg\varphi$ follows from Γ.

The second rule, for the elimination of negation, indicates that if a formula φ, as well as its negation, $\neg\varphi$, follow from the same set of premises Γ, then, from Γ also follow a contradiction, $\bot$. A set Γ from which follow a contradiction is called *inconsistent*.

The third rule indicates that, if Γ is an inconsistent set of formulae, then any formula φ follow from Γ.

There is no rule for the introduction of $\bot$ (or, the rule for the elimination of negation can be considered also as being the rule for the introduction of $\bot$).

Example 114. *Let us prove that the sequence* $\{P(a)\} \vdash \neg\neg P(a)$ *is valid:*

1. $\{P(a), \neg P(a)\} \vdash P(a)$; *(Hypothesis)*
2. $\{P(a), \neg P(a)\} \vdash \neg P(a)$; *(Hypothesis)*
3. $\{P(a), \neg P(a)\} \vdash \bot$; *($\neg e$, 1, 2)*
4. $\{P(a)\} \vdash \neg\neg P(a)$. *($\neg i$, 3)*

Example 115. *Let us prove that the sequence* $\{P(a), \neg P(a)\} \vdash P(b)$ *is valid:*

1. $\{P(a), \neg P(a)\} \vdash P(a)$; *(Hypothesis)*
2. $\{P(a), \neg P(a)\} \vdash \neg P(a)$; *(Hypothesis)*
3. $\{P(a), \neg P(a)\} \vdash \bot$; *($\neg e$, 1, 2)*
4. $\{P(a), \neg P(a)\} \vdash P(b)$. *($\bot e$, 3)*

Elimination of double negation

In the case of propositional logic, we also saw the following rule for the elimination of the double negation:

$$\neg\neg e \ \frac{\Gamma \vdash \neg\neg\varphi}{\Gamma \vdash \varphi}$$

Example 116. *Let us prove that the sequence* $\{(\neg P(a) \to Q(a)), \neg Q(a)\} \vdash P(a)$ *is valid:*

1. $\{(\neg P(a) \to Q(a)), \neg Q(a), \neg P(a)\} \vdash \neg P(a)$; *(Hypothesis)*
2. $\{(\neg P(a) \to Q(a)), \neg Q(a), \neg P(a)\} \vdash (\neg P(a) \to Q(a))$; *(Hypothesis)*
3. $\{(\neg P(a) \to Q(a)), \neg Q(a), \neg P(a)\} \vdash Q(a)$; *($\to e$, 2, 1)*
4. $\{(\neg P(a) \to Q(a)), \neg Q(a), \neg P(a)\} \vdash \neg Q(a)$; *(Hypothesis)*
5. $\{(\neg P(a) \to Q(a)), \neg Q(a), \neg P(a)\} \vdash \bot$; *($\neg i$, 4, 3)*
6. $\{(\neg P(a) \to Q(a)), \neg Q(a)\} \vdash \neg\neg P(a)$; *($\neg i$, 5)*
7. $\{(\neg P(a) \to Q(a)), \neg Q(a)\} \vdash P(a)$. *($\neg\neg e$, 6)*

Example 117. *Let us prove that the sequence* $\{\} \vdash (\mathtt{P}(\mathtt{a}) \vee \neg\mathtt{P}(\mathtt{a}))$ *is valid:*

1. $\{\neg(\mathtt{P}(\mathtt{a}) \vee \neg\mathtt{P}(\mathtt{a})), \mathtt{P}(\mathtt{a})\} \vdash \neg(\mathtt{P}(\mathtt{a}) \vee \neg\mathtt{P}(\mathtt{a}))$; *(Hypothesis)*
2. $\{\neg(\mathtt{P}(\mathtt{a}) \vee \neg\mathtt{P}(\mathtt{a})), \mathtt{P}(\mathtt{a})\} \vdash \mathtt{P}(\mathtt{a})$; *(Hypothesis)*
3. $\{\neg(\mathtt{P}(\mathtt{a}) \vee \neg\mathtt{P}(\mathtt{a})), \mathtt{P}(\mathtt{a})\} \vdash (\mathtt{P}(\mathtt{a}) \vee \neg\mathtt{P}(\mathtt{a}))$; *($\vee i_1$, 2)*
4. $\{\neg(\mathtt{P}(\mathtt{a}) \vee \neg\mathtt{P}(\mathtt{a})), \mathtt{P}(\mathtt{a})\} \vdash \bot$; *($\neg e$, 1, 3)*
5. $\{\neg(\mathtt{P}(\mathtt{a}) \vee \neg\mathtt{P}(\mathtt{a}))\} \vdash \neg\mathtt{P}(\mathtt{a})$; *($\neg i$, 4)*
6. $\{\neg(\mathtt{P}(\mathtt{a}) \vee \neg\mathtt{P}(\mathtt{a}))\} \vdash (\mathtt{P}(\mathtt{a}) \vee \neg\mathtt{P}(\mathtt{a}))$; *($\vee i_2$, 5)*
7. $\{\neg(\mathtt{P}(\mathtt{a}) \vee \neg\mathtt{P}(\mathtt{a}))\} \vdash \neg(\mathtt{P}(\mathtt{a}) \vee \neg\mathtt{P}(\mathtt{a}))$; *(Hypothesis)*
8. $\{\neg(\mathtt{P}(\mathtt{a}) \vee \neg\mathtt{P}(\mathtt{a}))\} \vdash \bot$; *($\neg e$, 7, 6)*
9. $\{\} \vdash \neg\neg(\mathtt{P}(\mathtt{a}) \vee \neg\mathtt{P}(\mathtt{a}))$; *($\neg i$, 8)*
10. $\{\} \vdash (\mathtt{P}(\mathtt{a}) \vee \neg\mathtt{P}(\mathtt{a}))$. *($\neg\neg e$, 9)*

5.6.5 Elimination of universal quantifier

The rule for the elimination of the universal quantifier is:

$$\forall e \; \frac{\Gamma \vdash (\forall x.\varphi)}{\Gamma \vdash \varphi[x \mapsto t]}$$

The elimination rule for the universal quantifier is quite simple: if we know that $(\forall x.\varphi)$ is a syntactic consequence from Γ, then we can instantiate the bound variable x with any term t.

Exercise 118. *Question: does the previous rule make sense if x does not appear in φ? For instance, from $(\forall x.\varphi)$ can we deduce $\Gamma \vdash \mathtt{P}(\mathtt{a})[\mathtt{x} \mapsto \mathtt{b}]$?*

Example 119. *Let us go back to an example previously discussed in which we have the two affirmations:* All men are mortal *and* Socrate is a man. *Can we conclude that* Socrate is mortal*? In order to answer to the question, we have to prove the sequence:*

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

where `Man` *and* `Mortal` *are predicates of arity 1 and* `s` *is a constant (functional symbol of arity 0) associated to the name Socrates. Here is the formal proof for the sequence:*

1. $\{\forall x.(\texttt{Man}(x) \rightarrow \texttt{Mortal}(x)), \texttt{Man}(s)\} \vdash \forall x.(\texttt{Man}(x) \rightarrow \texttt{Mortal}(x))$ (Hyp)
2. $\{\forall x.(\texttt{Man}(x) \rightarrow \texttt{Mortal}(x)), \texttt{Man}(s)\} \vdash (\texttt{Man}(s) \rightarrow \texttt{Mortal}(s))$ $(\forall e, 1, s)$
3. $\{\forall x.(\texttt{Man}(x) \rightarrow \texttt{Mortal}(x)), \texttt{Man}(s)\} \vdash \texttt{Man}(s)$ (Hyp)
4. $\{\forall x.(\texttt{Man}(x) \rightarrow \texttt{Mortal}(x)), \texttt{Man}(s)\} \vdash \texttt{Mortal}(s)$ $(\rightarrow e, 2, 3)$

Note that at step 2 of the proof, we used the rule $\forall e$ *which instantiates in the formula* $\forall x.(\texttt{Man}(x) \rightarrow \texttt{Mortal}(x))$ *the bound variable* x *with* s*:* $(\texttt{Man}(s) \rightarrow \texttt{Mortal}(s))$*. In natural language, this is similar with deducing by reasoning that* If Socrates is a man, then he is mortal *from* All men are mortal.

5.6.6 Introduction of existential quantifier

There is a duality of rules for the introduction and the elimination of quantifiers in the sense that the rule for introducing the existential quantifier from below can be seen as a dual rule for the elimination of universal quantifier:

$$\exists i \ \frac{\Gamma \vdash \varphi[x \mapsto t]}{\Gamma \vdash (\exists x.\varphi)}$$

The rule indicates that we can deduce $(\exists x.\varphi)$ when $\varphi[x \mapsto t]$ is a semantical consequence from Γ. Informally, if there is a concrete x — namely t — such that $\varphi[x \mapsto t]$ is true, we conclude that $(\exists x.\varphi)$ is true.

Example 120. *Let us prove that the sequence* $\{\texttt{P}(a)\} \vdash \exists x.\texttt{P}(x)$ *is valid:*

1. $\{\texttt{P}(a)\} \vdash \texttt{P}(a)$ (Hypothesis)
2. $\{\texttt{P}(a)\} \vdash \exists x.\texttt{P}(x)$ $(\exists i, 1)$

Note that in this case φ *is* $\texttt{P}(x)$ *and* $\varphi[\texttt{x} \mapsto \texttt{a}]$ *is* $\texttt{P}(x)[x \mapsto a]$*.*

Example 121. *Let us prove that the sequence* $\{\forall x.(\texttt{P}(x) \rightarrow \texttt{Q}(x)), \texttt{P}(a)\} \vdash \exists x.\texttt{Q}(x)$ *is valid:*

1. $\{\forall x.(\texttt{P}(x) \rightarrow \texttt{Q}(x)), \texttt{P}(a)\} \vdash \forall x.(\texttt{P}(x) \rightarrow \texttt{Q}(x))$ (Hypothesis)
2. $\{\forall x.(\texttt{P}(x) \rightarrow \texttt{Q}(x)), \texttt{P}(a)\} \vdash \texttt{P}(a)$ (Hypothesis)
3. $\{\forall x.(\texttt{P}(x) \rightarrow \texttt{Q}(x)), \texttt{P}(a)\} \vdash (\texttt{P}(a) \rightarrow \texttt{Q}(a))$ $(\forall e, 1, a)$
4. $\{\forall x.(\texttt{P}(x) \rightarrow \texttt{Q}(x)), \texttt{P}(a)\} \vdash \texttt{Q}(a)$ $(\rightarrow e, 3, 2)$
5. $\{\forall x.(\texttt{P}(x) \rightarrow \texttt{Q}(x)), \texttt{P}(a)\} \vdash \exists x.\texttt{Q}(x)$ $(\exists i, 4)$

5.6.7 Introduction of universal quantifier.

The rule for the introduction of the universal quantifier is:

$$\forall i \ \frac{\Gamma \vdash \varphi[x \mapsto x_0]}{\Gamma \vdash (\forall x.\varphi)} \ x_0 \notin vars(\Gamma, \varphi)$$

The rule from above says that we can conclude $\Gamma \vdash \forall x.\varphi$ if we first prove that $\varphi[x \mapsto x_0]$ is a syntactical consequence from Γ, where x_0 is a *new* variable: it does not appear in other formulae and we make no assumption over it.

Example 122. *Let us prove that the sequence* $\{\forall x.(\mathtt{P}(x) \to \mathtt{Q}(x)), \forall x.\mathtt{P}(x)\} \vdash \forall x.\mathtt{Q}(x)$ *is valid:*

1. $\{\forall x.(\mathtt{P}(x) \to \mathtt{Q}(x)), \forall x.\mathtt{P}(x)\} \vdash \forall x.(\mathtt{P}(x) \to \mathtt{Q}(x))$ (Hypothesis)
2. $\{\forall x.(\mathtt{P}(x) \to \mathtt{Q}(x)), \forall x.\mathtt{P}(x)\} \vdash \forall x.\mathtt{P}(x)$ (Hypothesis)
3. $\{\forall x.(\mathtt{P}(x) \to \mathtt{Q}(x)), \forall x.\mathtt{P}(x)\} \vdash (\mathtt{P}(x_0) \to \mathtt{Q}(x_0))$ $(\forall e, 1, x_0)$
4. $\{\forall x.(\mathtt{P}(x) \to \mathtt{Q}(x)), \forall x.\mathtt{P}(x)\} \vdash \mathtt{P}(x_0)$ $(\forall e, 2, x_0)$
5. $\{\forall x.(\mathtt{P}(x) \to \mathtt{Q}(x)), \forall x.\mathtt{P}(x)\} \vdash \mathtt{Q}(x_0)$ $(\to e, 3, 4)$
6. $\{\forall x.(\mathtt{P}(x) \to \mathtt{Q}(x)), \forall x.\mathtt{P}(x)\} \vdash \forall x.\mathtt{Q}(x)$ $(\forall i, 5)$

Note that for the sequences 3, 4 and 5, we use the variable x_0 *about which we make no assumption. Therefore, intuitively,* $\mathtt{Q}(x_0)$ *holds for any* x_0.

Exercise 123. *Prove that the following sequences are valid:*

1. $\{\forall x.(\mathtt{P}(x) \wedge \mathtt{Q}(x))\} \vdash \forall x.\mathtt{P}(x)$;
2. $\{\forall x.\mathtt{Q}(x), \mathtt{P}(a)\} \vdash \mathtt{P}(a) \wedge \mathtt{Q}(a)$;
3. $\{\forall x.\mathtt{P}(x), \forall x.\mathtt{Q}(x)\} \vdash \forall x.(\mathtt{P}(x) \wedge \mathtt{Q}(x))$.

5.6.8 Elimination of existential quantifier

The rule for the elimination of the existential quantifier is the following:

$$\exists e \ \frac{\Gamma \vdash (\exists x.\varphi) \qquad \Gamma \cup \{\varphi[x \mapsto x_0]\} \vdash \psi}{\Gamma \vdash \psi} \ x_0 \notin vars(\Gamma, \varphi, \psi)$$

The first hypothesis of the rule is $\Gamma \vdash (\exists x.\varphi)$, which, intuitively, ensures us that there is at least one term (they can be several) that can replace x

such that φ is a syntactical consequence from Γ. However, we don't know which are this terms (in the case they are several). We only know that there is at least one and we call it x_0. In order to prove the conclusion, that ψ is a syntactical consequence from Γ, we have to analyze several cases for x_0. This is summarized by the second hypothesis of the rule, where we have to prove that ψ is a syntactical consequence from $\Gamma \cup \{\varphi[x \mapsto x_0]\}$.

Example 124. *Let us prove that the sequence* $\{\forall x.(P(x) \to Q(x)), \exists x.P(x)\} \vdash \exists x.Q(x)$ *is valid:*

1. $\{\forall x.(P(x) \to Q(x)), \exists x.P(x)\} \vdash \exists x.P(x)$ (Hypothesis)
2. $\{\forall x.(P(x) \to Q(x)), \exists x.P(x), P(x_0)\} \vdash P(x_0)$ (Hypothesis)
3. $\{\forall x.(P(x) \to Q(x)), \exists x.P(x), P(x_0)\} \vdash \forall x.(P(x) \to Q(x))$ (Hypothesis)
4. $\{\forall x.(P(x) \to Q(x)), \exists x.P(x), P(x_0)\} \vdash (P(x_0) \to Q(x_0))$ $(\forall e, 3, x_0)$
5. $\{\forall x.(P(x) \to Q(x)), \exists x.P(x), P(x_0)\} \vdash Q(x_0)$ $(\to e, 4, 2)$
6. $\{\forall x.(P(x) \to Q(x)), \exists x.P(x), P(x_0)\} \vdash \exists x.Q(x)$ $(\exists i, 5)$
7. $\{\forall x.(P(x) \to Q(x)), \exists x.P(x)\} \vdash \exists x.Q(x)$ $(\exists e, 1, 6)$

Note that in order to prove the 7th sequence, we used sequences 1 and 6. The former one was proved by steps 2,3,4 and 5, where we also used as a hypothesis the formula $P(x_0) (= P(x)[x \mapsto x_0])$.

5.6.9 Other rules

Another useful rule, that does not necessary corresponds to some operator, is the extension rule, that was also presented in the case of natural deduction for propositional logic:

$$\text{Extension } \frac{\Gamma \vdash \varphi}{\Gamma, \varphi' \vdash \varphi}$$

This rule indicates the fact that, if φ is a consequence from a set of formulas Γ, then φ is also a consequence of $\Gamma \cup \{\varphi'\}$ (for any φ'). In other words, we can extend the set of premises of a valid sequence and we get another valid sequence.

Example 125. *Here is a proof of* $\{P(a), \neg Q(a), P(f(a)), (P(b) \wedge Q(b))\} \vdash \neg\neg P(a)$*:*

1. $\{P(a), \neg P(a)\} \vdash P(a)$; *(Hypothesis)*

2. $\{P(a), \neg P(a)\} \vdash \neg P(a)$; *(Hypothesis)*

3. $\{P(a), \neg P(a)\} \vdash \bot$; *($\neg e$, 1, 2)*

4. $\{P(a)\} \vdash \neg\neg P(a)$; *($\neg i$, 3)*

5. $\{P(a), \neg Q(a)\} \vdash \neg\neg P(a)$; *(Extindere, 4)*

6. $\{P(a), \neg Q(a), P(f(a))\} \vdash \neg\neg P(a)$; *(Extindere, 5)*

7. $\{P(a), \neg Q(a), P(f(a)), (P(b) \wedge Q(b))\} \vdash \neg\neg P(a)$. *(Extindere, 6)*

5.7 Natural deduction system

The natural deduction for first order logic is the deductive system formed by all rules presented in the previous section. Here is the sum up of all rules:

$$\wedge i\ \frac{\Gamma \vdash \varphi_1 \qquad \Gamma \vdash \varphi_2}{\Gamma \vdash (\varphi_1 \wedge \varphi_2),} \qquad \wedge e_1\ \frac{\Gamma \vdash (\varphi_1 \wedge \varphi_2)}{\Gamma \vdash \varphi_1,}$$

$$\wedge e_2\ \frac{\Gamma \vdash (\varphi_1 \wedge \varphi_2)}{\Gamma \vdash \varphi_2,} \qquad \to e\ \frac{\Gamma \vdash (\varphi_1 \to \varphi_2) \qquad \Gamma \vdash \varphi_1}{\Gamma \vdash \varphi_2,}$$

$$\to i\ \frac{\Gamma, \varphi_1 \vdash \varphi_2}{\Gamma \vdash (\varphi_1 \to \varphi_2),} \qquad \vee i_1\ \frac{\Gamma \vdash \varphi_1}{\Gamma \vdash (\varphi_1 \vee \varphi_2),} \qquad \vee i_2\ \frac{\Gamma \vdash \varphi_2}{\Gamma \vdash (\varphi_1 \vee \varphi_2),}$$

$$\vee e\ \frac{\Gamma \vdash (\varphi_1 \vee \varphi_2) \qquad \Gamma, \varphi_1 \vdash \varphi' \qquad \Gamma, \varphi_2 \vdash \varphi'}{\Gamma \vdash \varphi',}$$

$$\neg e\ \frac{\Gamma \vdash \varphi \qquad \Gamma \vdash \neg\varphi}{\Gamma \vdash \bot,} \qquad \neg i\ \frac{\Gamma, \varphi \vdash \bot}{\Gamma \vdash \neg\varphi,} \qquad \bot e\ \frac{\Gamma \vdash \bot}{\Gamma \vdash \varphi,}$$

$$\textsc{Hypothesis}\ \frac{}{\Gamma \vdash \varphi}\ \varphi \in \Gamma, \qquad \textsc{Extindere}\ \frac{\Gamma \vdash \varphi}{\Gamma, \varphi' \vdash \varphi,} \qquad \neg\neg e\ \frac{\Gamma \vdash \neg\neg\varphi}{\Gamma \vdash \varphi.}$$

$$\forall e\ \frac{\Gamma \vdash (\forall x.\varphi)}{\Gamma \vdash \varphi[x \mapsto t]} \qquad \exists i\ \frac{\Gamma \vdash \varphi[x \mapsto t]}{\Gamma \vdash (\exists x.\varphi)}$$

$$\forall i\ \frac{\Gamma \vdash \varphi[x \mapsto x_0]}{\Gamma \vdash (\forall x.\varphi)}\ x_0 \notin vars(\Gamma, \varphi)$$

$$\exists e\ \frac{\Gamma \vdash (\exists x.\varphi) \qquad \Gamma \cup \{\varphi[x \mapsto x_0]\} \vdash \psi}{\Gamma \vdash \psi}\ x_0 \notin vars(\Gamma, \varphi, \psi)$$

In proofs we are allowed to use derived rules (including the ones presented for propositional logic).

5.8 Soundness and Completeness of Natural Deduction for Forst Order Logic

Theorem 126 (Soundness for natural deduction)**.** *For any set* Γ *of formulae and any formula* φ*, if the sequence* $\Gamma \vdash \varphi$ *is valid, then* $\Gamma \models \varphi$*.*

Exercise 127. *Prove Theorem 126.*

Theorem 128 (Completeness for natural deduction)**.** *For any set* Γ *of formulae and any formula* φ*, if* $\Gamma \models \varphi$ *then the sequence* $\Gamma \vdash \varphi$ *is valid.*

The proof of the completeness theorem exceeds the level of this course.

Remark. *Note that, using the two soundness and completeness theorems, the relation $\vdash$ coincides with $\models$, even if they have different meanings.*

5.9 Exercises

Exercise 129. *Prove that the next sequences are valid:*

1. $\{((P(a) \wedge Q(a)) \wedge \forall x.P(x))\} \vdash (Q(a) \wedge \forall x.P(x))$;
2. $\{((P(a) \wedge Q(a)) \wedge \forall x.P(x)), \forall x.Q(x)\} \vdash (\forall x.Q(x) \wedge Q(a))$;
3. $\{((P(a) \wedge Q(a)) \wedge \forall x.P(x))\} \vdash (\forall x.P(x) \wedge (Q(a) \wedge P(a)))$;
4. $\{((P(a) \wedge Q(a)) \rightarrow \forall x.P(x)), P(a), Q(a)\} \vdash \forall x.P(x)$;
5. $\{(P(a) \rightarrow \forall x.P(x)), P(a), Q(a)\} \vdash (Q(a) \wedge \forall x.P(x))$;
6. $\{(P(a) \rightarrow P(b)), (Q(a) \rightarrow P(b))\} \vdash ((P(a) \vee Q(a)) \rightarrow P(b))$;
7. $\{\neg(P(a) \wedge Q(a))\} \vdash (\neg P(a) \vee \neg Q(a))$;
8. $\{\neg(\neg P(a) \vee \neg Q(a))\} \vdash (P(a) \wedge Q(a))$;
9. $\{\neg(\neg P(a) \wedge \neg Q(a))\} \vdash (P(a) \vee Q(a))$;

Exercise 130. *Which of the following sequences are valid?*

1. $\{\forall x.(P(x) \wedge Q(x))\} \vdash \forall x.P(x)$;
2. $\{\forall x.Q(x), P(a)\} \vdash (P(a) \wedge Q(a))$;
3. $\{\forall x.P(x), \forall x.Q(x)\} \vdash \forall x.(P(x) \wedge Q(x))$;
4. $\{\exists x.\exists y.P(x, y)\} \vdash \exists y.\exists x.P(x, y)$;
5. $\{\exists x.\forall y.P(x, y)\} \vdash \forall y.\exists x.P(x, y)$; *What about* $\{\forall y.\exists x.P(x, y)\} \vdash \exists x.\forall y.P(x, y)$*?*
6. $\{\neg(\exists x.P(x))\} \vdash \forall x.\neg P(x)$;
7. $\{\forall x.\neg P(x)\} \vdash \neg(\exists x.P(x))$;