Logic for knowledge representation, learning, and inference

Luciano Serafini

Contents

CHAPTER 1

First Order Logic

1. Introduction

First-order logic can be understood as an extension of propositional logic. In propositional logic the atomic formulas have no internal structure—they are propositional variables that are either true or false. In First-order logic the atomic formulas are predicates that assert a property of an element of the domain of interest or a nothinrelationship among certain elements. To emphasize this fact First Order Logic is also called *predicate logic*. First order logic, though not as expressive as full spoken language such as English or Italian, introduces a more structured syntax to express propositions. In particular it allows to express the fact that a certain object has some property by introducing symbols for objects and symbols for properties, and a way to combine the two in order to obtain the proposition that statss that the property holds for the object. FOL not only allow to state property of objects but also relationships between objects as well as functions on objects. Form the forst order logic prespective a "world" is described as a set of objects (aka domain, or universe) and a set of funcitons and relations on these objects.

To understand what we gain by using FOL w.r.t .propositional logic consider this simple example.

EXAMPLE 1.1. *Suppose that we want to express the four propositions expressed by the following four englishg statements:*

1. *Mary is a person*
2. *John is a person*
3. *Mary is mortal*
4. *Mary and John are siblings*

Since the four statements expresses four differnt propositions, in propositional logic we have to introduce four different propositional variables, say p, q, r, and s, with the following intuitive menaing.

- *p that stands for Mary is a person*
- *q that stands for John is a person*
- *r that stands for Mary is mortal*
- *s that stands for Mary and John are siblings*

By doing so, we loose the many information about the relationship of the four proposition. For instance we don't represent the fact that proposition (1) and (3) are about the same person, namely Mary; that proposition (1) and (2) states the same property about two individual; that proposition (4) states a relationship about the two individuals which are involved in proposition (1) and (2). All this relationship between propositions are lost since propositional variables are atomic, i.e., they are not build of simpler components.

The langauge of FOL instead introduces symbols to describe the entities and symbols for describing properties. For instance two symbols, e.g., M and J are introduced in order to denote the two entities John and Mary, and a symbol P is introduced to denote the property of being a person. Using these symbols in FOL we can build the formula $P(M)$ *that represents the proposition that "the property P holds for the entity M" and the formula* $P(J)$ *that represents the proposition "the property P holds for the entity J" In a similar fashon FOL introduces a symbol S to represent the sibling relation and uses the formula* $S(J, M)$ *to state the property that John and Mary are siblings.*

Entities of a domain can be specified by providing a direct name, e.g., John, Mary, …... but we can also refer to an entity in terms on how it relates with another entity. For instance, even if we don't have a specific symbol to denot the father of Mary, in natural langauge we can refer to him with the *definite description* "the father of Mary". In this case "the father of" is a construct that allows to build the description of an entity starting from a description of another entity. We can apply this construction as many time as we want, for instance obtaining definite descriptions such as "the father of the father of Mary". The language of first order logic introduces symbols to denote these constructors of description, they are called *functional symbols*. For instance if F is a functional symbol corresponding to the definite description constructor "the father of", in FOL we can build the formulas $P(F(A))$ that states that the father of Alice is a Person.

Not only FOL allows to have expressions to denote specific objects, FOL also allows a set of symbols to denot any (not specific) object. This is similar to the work "object" or "entity" or "thing" or "element" in english. This are very generic terms that can be used to denote any specific element. They are different from proper names since by composing a proer name with a predicate you obtain a proposition, like in "John is a person", instead by composing a variables with a predicate you obtain "the object is a person" which is not a proposition, since we don't know to which objects it refers to. Individual variables are important since they are necessary to introduce quantification. This is similar to what happens in englis when we use "somebody" or "everybody", and "something" or "everything" Every individual variable x can be *quantified* either universlly (like in "everybody") or existentially (like in somebody) So we can say "somebody is a person" or "everybody is a person" or "every obejct is a person". For this purpose the language of FOL provides two new logical symbols (in addition to the boolean connectives) called *quantifiers* $\forall x$ and $\exists x$ for every individual variable x that stands for "for all individual" and "there is an indivudual".

In summary, while propositional logic describe a world in terms of a set of propositions which are true and false, in FOL describe the worldss in terms of a set of objects, which constitute the so called *interpretation domain* and a set of properties and relationship between them.

2. Syntax of FOL

The language of FOL logic is defined relative to a signature. A signature Σ consists of a set of constant symbols $c_1, c_2, \dots$, a set of function symbols $f_1, f_2, \dots$ and a set of predicate symbols $p_1, p_2, \dots$. Each function and predicate symbol has an arity $k > 0$. We will often refer to predicates as relations. Predicates and function with arity equal to k are called k-ary predicate and k-ary functions.

We also suppose that Σ contains an infinite set of variables $x_0, x_1, \ldots$. To denote variables we also use the last letters of the alphabet x,y, ..., possibly with indices. For constants instead we use the first letters of the alphabet a, b, ... possibly with indices.

Not every seq1uence of symbols in Σ are legal expressioons. In FOL we provide a grammar for two types of expressions called *terms*, which are used to denote objects in the domain of interests, and *formulas*, which are intended to denote propositions about the domain of interests, which can be either true or false.

2.1. Terms in FOL. Terms are expression built starting from constants and individual variables, by composing them with funciton symbols. As clarified above they are descriptions of the objects of the domain of interest. We provide an inductive definition of the set of terms for a signaturee Σ.

DEFINITION 1.1. *Given a signature Σ, the set of terms (more precisely Σ-terms) is defined by the following set:*

- *a constant c_i is a term*
- *a variable x_i is a term*
- *if $t_1, \ldots, t_k$ are terms and f is a k-ary function symbol, then $f(t_1, \ldots, t_k)$ is a term*
- *nothing else is a term.*

Intuitively terms are used to denote objects in the domain of the world we want to describe with our FOL language

EXAMPLE 1.2 (Terms).

- x_i*: denotes a generic object of the domain; Variables are supposed to range over all the objects of the domain and they can be instantiated with any one of them. This is why they are also referred as "individual variables", to keep them distinguished from "propositional variables" which instead ranges over propositions.*
- c_i*; a constant denotes one specific element of the domain.*
- $f_i(x_j, c_k)$; *complex term. This is similar to what we say in English when we refer to a person in terms of "functions" of another preson, e.g., "the father of John", the person that stands between John and Mary*
- $f(g(x, y), h(x, y, z), y)$*; a more complex terms. e.g. the father of the person that stands between John and Mary*

A term is *ground* or *closed* if it does not contain individual variables. Ground terms are descriptions of some specific object in the domain. If there are no constant symbol in the siguature then there is no ground terms. If there are no function symbol then the set of ground terms coincides with the set of constants. If we have at least one constant and one function symbol then the set of ground terms is infinite. For instance if we have the constant Mary and the function motherOf then we have the following infinite set of terms:

$$\begin{aligned} &\mathsf{Mary} \\ &\mathsf{motherOf(Mary)} \\ &\mathsf{motherOf(motherOf(Mary))} \\ &\mathsf{motherOf(motherOf(motherOf(Mary)))} \\ &\dots \end{aligned}$$

2.2. Formulas in FOL. The simplest FOL expression that specifies a proposition is called *atomic formula* and it states that a relation espressed by an n-ary predicate[1] p holds for an n-tuple of objects specified by the terms t_1, t_2,..., t_n.

Definition 1.2 (Atomic formula). *An atomic formula on a signature Σ is an expression of the form $p(t_1, \dots, t_n)$ there p is an n-ary predicate of Σ and t_i are Σ terms.*

Roughly speaking atomic formulas in FOL correspond to propositional variables in propositional logic. We will make this correspondence more precise later. For now it is enough to think that the atomic formula $Happy(John)$ (where "John" is a constant symbol and "Happy" is a 1-ary (unary) predicate, corresponds to a propositional variables that we would have introduced in propositional logic to formalize the proposition that John is happy. In fact, sometimes in propositional logic we introduce the propositional variable "Happy(John)", but in spite of the fact that it looks the same as the first order atomic formula, it should be considered as a string and not as a structured object.

As in propositional logic complex formulas can be built starting from atomic formulas;

Definition 1.3 (First order formulas). *A (first order) formula on a signature Σ is defined as follows:*

- *an atomic formula is a formula;*
- *if ϕ and ψ are formulas then $\neg\phi$, $\phi \wedge \psi$, $\phi \vee \psi$, $\phi \rightarrow \psi$, and $\phi \leftrightarrow \psi$ are formulas;*
- *if ϕ is a formula and x and individual variable, the $\forall x.\phi$ and $\exists x.\phi$ are formulas;*
- *Nothing else is a formula.*

Notice that the definition of first order logic formula extends the definition of formula in propositional logic by adding a new rules for the operator $\forall x$ and $\exists x$. These operators are called *universal* and *existential* quantifier, respectively . The intuitive reading fo $\forall x.\phi$ is "for every x ϕ" and the intuitive reading of $\exists x \phi$ is "there is an x such as ϕ.

Example 1.3. *Consider a signature with a single 2-ary (binary) relation symbol R. Since there are no constant symbols or function symbols, the only terms are variables x_1, x_2, ... The set of atomic formulas contains $R(x, x)$, $R(x, y)$,*

[1]An n-ary predicate, where n is a natural number, is a predicate with arity equal to n a predicate with arity equal to n.

$R(x_i, x_j)$ *Examples of formulas are*

$$\forall x.\forall y.\forall z.(R(x,y) \wedge R(y,z) \rightarrow R(x,z))$$
$$\forall x, \forall y.\exists z.(R(x,y) \rightarrow R(x,z) \wedge R(z,y)$$

This formulas expresse that R is a transitive relation and a dense relation, respectively.

Consider a signature with a constant symbol 0, unary function symbol s, and unary predicate symbol E. Terms over this signature include x, 0, $s(0)$, $s(x)$, $s(s(0))$, $s(s(s \dots s(0)$ $s(s(s \dots s(x) \dots))$ .... Atomic formulas are obtained by applying the unary predicate E to these terms, e.g. $E(x)$, $E(0)$, $E(s(0))$, $E(s(x))$, $E(s(s(0)))$, $E(s(s(s \dots s(0))$, $E(s(s(s \dots s(x) \dots)))$ .... An example of (non atomic) formula is the following:

$$E(0) \wedge \forall x(E(x) \leftrightarrow \neg E(s(x))$$

Sometimes we write function symbols and predicate symbols infix to improve readability:

Example 1.4. *Consider a signature with a constant symbol* 1*, binary function symbol* $+$*, and a binary relation symbol* $<$*, both written infix. Then* $x+1$ *is a term (that is the infix notation of* $+(x,1)$*) and* $\forall x.(x < (y+1))$ *is a formula (corresponding to the infix notation of* $\forall x.(<(x, +(x,1)))$*).*

As form propositional logic we should also clarify in case of ambiguity (because we miss the parentesis) what are the precedence of the quantifier w.r.t, connectives. We have that $\forall x$ and $\exists x$ have the same priority of $\neg$. This implies that they should be applied before all the other connectives.

Quantifiers add expressive power to first order logic w.r.t propositional logic, as they allow to state in a single formula a fact that holds for all the induvidual of the domain, without eplicitly enumerating the property of each of them. Notice that this is possible also when the domain contains an infinite set of objects (e.g., the natural numbers). In this case it would be impossible to write a propositional formula since it would have infinite length.

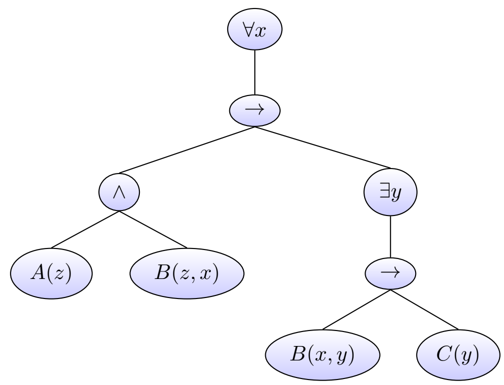
Example 1.5. *Consider a signature with a constant symbol* 0*, unary function symbol* s*, and unary predicate symbol* E*. As we have seen in the previous example with this signature we can build an infinite set of terms, wich might denote an infinite set of objects (e.g, the natural numbers). To express the same fact of the previous example, without the help of universal quantifier we should have written an infinite formula*

$$E(0) \wedge (E(0) \leftrightarrow \neg E(s(0))) \wedge (E(s(0)) \leftrightarrow \neg E(s(s(0)))) \wedge (E(s(s(0))) \leftrightarrow \neg E(s(s(s(0))))) \wedge \dots$$

which is not possible in propositional logic.

For every first order formula ϕ we can define the formula tree that describe the structure of the formula. The formula tree expresses how the formula has been built using the rules of Definition 1.3. The definition is analogous to the formula tree for propositional formulas, with the addition of the universal and existential quantifier.

Example 1.6. *The tree of the formula* $\forall x(A(c) \wedge B(c,x) \rightarrow \exists y(B(x,y) \vee C(y)))$



2.3. Free and bound variables. If $\forall x\psi$ (resp. $\exists x\psi$) is a subformula of a formula ϕ, we way that ϕ is *the scope* of the that specific occurrenc of the quantifier $\forall x$ (resp. $\exists x$). Notice that the scope is defined for every occurrence of a quantifier, and that different occurrences of the same quantifier have different scope. For instance the scope of the forst occurrence of $\forall x$ in $\forall x P(x) \land \forall x(Q(x) \land \exists y.R(y))$ is $P(x)$, and the scope of the second occurrence of $\forall x$, is $Q(x) \land \exists y.R(y))$.

The notion of scope of the occurrence of a quantifier becomes rather clear if you think to the formula tree of a formula ϕ. Indeed an occurrence of a quantifier $\forall x$ of a formula ϕ corresponds to a node n of the tree of the formula ϕ labelled with $\forall x$; its scope is the sub-tree rooted at the only child of n.

Definition 1.4. *The occurrence of a variable x in a formula ϕ is* free *if it does not occour in the scope of a quantifier $\forall x$ or $\exists x$. A variable x is free in ϕ if there is at least one occurrence of x in ϕ that is free.*

Proposition 1.1. *Let $FV(\phi)$ denotes the set of free variables of ϕ, then*

(1) $FV(P(t_1, \ldots, t_n))$ is the set of variables that occour in some t_i;
(2) $FV(\neg\phi) = FV(\phi)$;
(3) $FV(\phi \circ \psi) = FV(\phi) \cup FV(\psi)$ for every connective $\circ \in \{\land, \lor, \to, \leftrightarrow\}$
(4) $FV(Qx.\phi) = -FV(\phi) \setminus \{x\}$ for $Q \in \{\forall, \exists\}$

If we want to make explicit the set of free variables of a formula ϕ, we write $\phi(x_1, \ldots, x_n)$ where $\mathrm{FV}(\phi) = \{x_1, \ldots, x_n)$. This is just notation; it should not interpreted as we add the expression $(x_1, \ldots, x_n)$ at the end of the formula. Actually, $\phi(x_1, \ldots, x_n)$ is the same formula as ϕ, they are two different way to denotes the same formula. In the second method we make explicit that in $\phi(x_1, \ldots, x_n)$ we made eplicit the set of free variables.

Example 1.7. *consider the formula*

$$\forall x(P(x, y) \land \exists z P(x, z) \to \exists x(Q(x, y, z, w))) \quad (1)$$

From the previous examples, it should be clear that a quantifier can occour also in the scope of another quantifier. We should clarify whaT happens when a quantifire of a variable x occours in the scope of another occurrence of a quantifier of the same wariable. Who has the precedence on the variable?

The occurrence of a variable x is *bounded* by a quantifier Qx (either $\forall x$ or $\exists x$) is the first node labelled with Qx, that occurs in the path from the leaf of the occurrence of x to the root of the formula tree. If there is no such a node we say that the occurrence of x is unbound or *free*.

EXAMPLE 1.8. *In the previous example, the occurrence of* x *in the nodes* $B(z,x)$ *and* $B(x,y)$ *is the root node* $\forall x$*; the occurrence of* y *in* $B(x,y)$ *is the node* $\exists y$*. Instead, the occurrences of* z *in* $A(z)$ *and* $B(z,y)$ *are unbound.*

An occurrence of a variable x in a formula ϕ is bound if that occurrence is bound by some quantifier. An occurrence that is not bound is said to be free. Note that different occurrences of the same variable in a given formula can be both bound and free, e.g., variable x occurs both bound and free in the formula $P(x) \wedge \exists x P(x)$. A formula with no free variables is said to be *closed* or a *sentence*. The formulas. If a formula ϕ contains a free occurrence of a variable x is denoted by $\phi(x)$; if ϕ contains at least one free occurrence for each variable $x_1 \dots, x_n$ we write $\phi(x_1, \dots, x_n)$.

EXAMPLE 1.9. *In the formula* $\forall x(Q(x,y) \rightarrow R(x,y))$ *the occurence of* y *is free while the occurence of* x *is bound, therefore* y *is free while* x *is bound. In the formula* $\forall x(Q(x,y) \rightarrow \exists y R(x,y))$ *the occurrence of* y *in* $Q(x,y)$ *is free while the occurence of* y *in* $R(x,y)$ *is bound. The two occurrences of* x *are bound. Therefore, the variable* x *is bound while the variable* y *is both free.*

2.4. Substitution of variables with terms. One of the most frequent operation that we have to do in a formula $\phi(x_1, \dots, x_n)$ that contains the free variables $x_1, \dots, x_n$ is to replace one, some, or all free variables with terms. This iperation intuitively means that you instantiate the variable, which intuitively denotes any element of the domain, with one specific element described by the corresponding term. The terms which replace variables can contains other variables, and as we will see later this might creeate some problem.

DEFINITION 1.5. *A* substitution σ *is a function that assigns to every individual variable* x *a term* t*. A substitution that replaces* $x_1, \dots, x_n$ *with* $t_1, \dots, t_n$ *respectively and leave every other variables unchanged is denoted by* $x_1/t_1, \dots, x_n/t_n$*. The result of applying the substitution* $x_1/t_1, \dots, x_n/t_n$ *to a term* $t(x_1, \dots, x_n)$ *or to a formula* $\phi(x_1, \dots, x_n)$*, denoted by* $t(x_1, \dots, x_n)[\sigma]$ *and* $\phi(x_1, \dots, x_n)[\sigma]$ *is the term or the formula obtained by replacing simultaneusly every free occurrence of every* x_i*, with* t_i *if this occurrence doees not occour in the scope of a quantifier of a variable of* t_i*.*

EXAMPLE 1.10. *Let* $\phi(x,y) = R(x,y) \rightarrow \exists z R(y,z)$ *and* $\sigma = [x/a, y/f(z)]$ *be a substitution. The application of* σ *to* ϕ *is* $\phi(a, f(z))$ *is* $R(a, f(z)) \rightarrow \exists z R(y,z)$*. Notice that the second occurrence of* y *in* ϕ *is not substituted because it occurs in the scope of the quantifier* $\exists z$ *and* z *occurs in the terms for which* y *is replaced.*

Some remarks are in order. The application of a substitution $x_1/t_1, \dots, x_n/t_n$ to a term or a formula, should be done simultaneusly. This means that if some t_i contains a variable x_j this variable should not be substitute. For instance the application of the substitution $x/a, y/x$ to $P(x,y)$, i.e., $P(x,y)[x/a, y/x]$ is $P(a,x)$ and not $P(a,a)$.

The second remarks concerns the condition given at the end of Definition 1.5. According to this condition, we cannot replace an occurrence of x with a term $t(y)$ if x occurs in the scope of a quantifier on y. For instance the substitution $x/f(y)$ in the formula $P(x,y) \wedge \exists y R(x,y)$, is equal to $P(f(y),y) \wedge \exists y R(x,y)$. Notice that the occurrence of y in the scope of the existential quantifier $\exists y$ is not affected by the substitution, since y occurs in $f(y)$. For an intuiition on why this restriction is important consider the following example

EXAMPLE 1.11. *To represent in FOL any person x does not like some piece of music*

$$\forall x(\textsf{\textit{person}}(x) \rightarrow \exists y(\textsf{\textit{music}}(y) \wedge \neg \textsf{\textit{likes}}(x, y))) \tag{2}$$

If we believe that (2) is true, then, intuitively, if we replace the variable x with any term t denoting some person, we should also believe in the result of the substitution. Suppose that the language contains the function symbols composer *such that* author(x) *denotes the composer of apiece of music x. Consider the substitution $\sigma = x/\textsf{\textit{composer}}(y)$. Notice that there is one occurrence of x in (2) that is in the scope of a quantifier of y, which is a variable in the term by which x is replaced. If in the application of σ to (2) we ignore the condition given at the end of Definition 1.5 we obtain the formula*

$$\exists y(\textsf{\textit{music}}(y) \wedge \neg \textsf{\textit{likes}}(\textsf{\textit{composer}}(y), y))$$

which is a closed formula that states there is some piece of music which does not like to his/her composer. However, intuitively, this does not logically follows from (2).

DEFINITION 1.6. *A term t is free for x in ψ if x does not occour in the scope of some quantifier of a variable of t.*

From now onw we will consider only substitutions $[x_i/t_1, \ldots, x_n/t_n]$ for which t_i is free for x_i in ϕ for every $1 \leq i \leq n$.

Finally, we introduce an additional notation, which is rather intuitive for expressing the result of the application of a substitution $x_1/t_1, \ldots, x_n/t_n$ to a term $t(x_1, \ldots, x_n)$ or to a formula $\phi(x_1, \ldots, x_n)$. When the contest is clear, we replace the long notations $t(x_1, \ldots, x_n)[x_1/t_1, \ldots, x_n/t_n]$ and $\phi(x_1, \ldots, x_n)[x_1/t_1, \ldots, x_n/t_n]$, with $t(t_1, \ldots, t_n)$ and $\phi(t_1, \ldots, t_n)$.

3. Semantics of first order logic

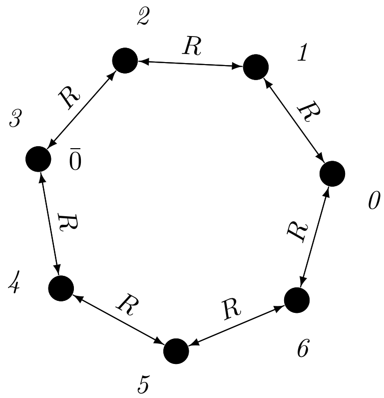
The semantics of a logic describes how the symbols and the expressions of such a logic can be interpreted in a mathematical structura that is intended to formalize a state of the world. The semantics of propositional language is just an assignment to propositional variables that state what is true and wat is false. The semantics of first order logic is more complex and is based on the notion of Σ-structure. The semantics of order logic with signature Σ is given in terms of a mathematical structure, called Σ-structure, or equivalently a Σ-interpretation.

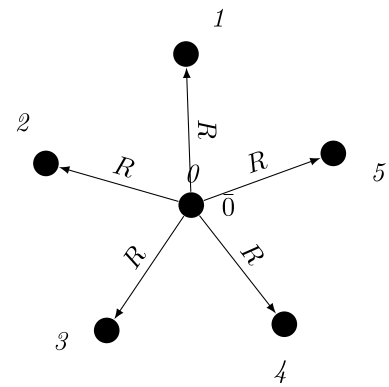
DEFINITION 1.7 (Σ-interpretation). *A Σ-interpretation $\mathcal{I}$, or an interpretation of Σ, consistes of:*

- *A non-empty set $\Delta_{\mathcal{I}}$ called the universe or the domain of the structure $\mathcal{I}$;*
- *for each constant symbol c, an element $c^{\mathcal{I}} \in \Delta_{\mathcal{I}}$;*
- *for each k-ary function symbol f in a k-ary function, $f^{\mathcal{I}} : \Delta_{\mathcal{I}}^k \longrightarrow \Delta_{\mathcal{I}}$;*
- *for each k-ary predicate symbol P in a k-ary relation $P^{\mathcal{I}} \subseteq \Delta_{\mathcal{I}}^k$*

where $\Delta_{\mathcal{I}}^k$ denotes the set of k-tuple of elements in $\Delta_{\mathcal{I}}$.

EXAMPLE 1.12. *Consider the signature $\Sigma = (\bar{0}, R)$ that contains only one cnstant symbol $\bar{0}$ and the binary predicate R Three examples of structures for this signature are shown in the following picture*

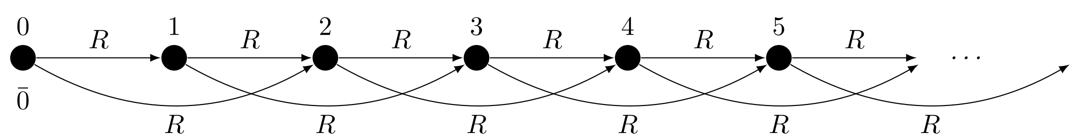




The domain of the Σ-structure on the left is equal to $\{0, 1, 2, 3, 4, 5, 6\}$ the constant $\bar{0}$ is interpreted in 3 and the predicate symbol R is interpreted in the set of pairs $\{(0, 1), (1, 0), (1, 2), (2, 1), \dots (5, 6), (6, 5), (6, 1), (1, 6)\}$. Notice the difference between $\bar{0}$ which is an element of Σ and the element 0 which is an element of the doain of the Σ-structure.

On the right we hav a Σ-structure which domain is $\{0, 1, 2, 3, 4, 5, 6, 7\}$ the constant $\bar{0}$ is interpreted in 0 and the prtedicate symbol R is interpreted in the set of pairs $\{(0, 1), (0, 2), \dots, (0, 7)\}$. The above example are finite *structures since the interpretation domain is finite, i.e., it conteins a finite number of elements.*

It is also possible to have infinite Σ-structures. For instance the following structure is infinite as it contains an infinite set of objects.



The domain of the above structure is the set of natural numbers $\mathbb{N} = \{0, 1, 2, 3, 4, \dots\}$, the constant $\bar{0}$ is interpreted in 0 and the relation R is the set of pairs $\{(n, n+1) \mid n \in \mathbb{N}\}$

Well studied structures in First order logic and in mathematics are $(\mathbb{N}, 0, <)$ wehre $\mathbb{N}$ is the set of natural numbers $<$ is the order relation between the natural numbers and 0 is the smalles natural number. This structure is very similar to the one shown above

EXAMPLE 1.13. *An undirected graph can be considered as a Σ-structure for the signature that contains the binary relation symbol E (for edge), where E is interpreted as the edge relation. For example, the graph shown in Figure1 can be represented by a structure $\mathcal{I}$ with universe $\Delta_{\mathcal{I}} = \{1, 2, 3, 4\}$ and $E^{\mathcal{I}}$ is the irreflexive symmetric binary relation*

$$E^{\mathcal{I}} = \{(1, 2), (2, 3), (3, 4), (4, 1), (2, 1), (3, 2), (4, 3), (1, 4), (1, 3), (3, 1)\}$$

EXAMPLE 1.14. *TODO: Add an example of a sigma structure that contains also constants and functions*

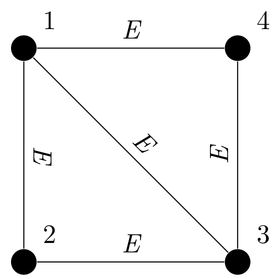


FIGURE 1.

The terms and the formulas that can be constructed from the signature Σ using the rules of First Order Logic are interpreted in a Σ-structure. However, a Σ-structure provides the "meaning" of all the symbols in Σ with the exception of the variables (which are not in Σ). Since terms and formulas contains also variable, in order to interpret terms and formulas in a Σ structure we also have to provide an assignment to the individual varialbes.

On could ask why the interpretation of individual variables is kept separated form the interpretation of the symbols in Σ. There is no technical reason, indeed to keed them separated, and in some textbook you can find that an interpretiaton also interpret the variabees. The main reason is that, in order to inverpret the unversa and existential quanfiers on a variable x we have to consider all (or some) instantiation of x, so we want to allow to modify the interpretation of variables in order to provide the meaning of the quantified formulas.

DEFINITION 1.8. *Given an interpretation $\mathcal{I}$ for the signature Σ (or equivalently a Σ-structure $\mathcal{I}$), an* assignment *is a function $a : X \to \Delta_{\mathcal{I}}$ form the set of individual variables X to the interpretation domain $\Delta_{\mathcal{I}}$.*

Given a Σ-structure $\mathcal{I}$ and an assignment a to the individual variables in $\Delta_{\mathcal{I}}$ we are now ready to define how terms and formulas are interpreted in $\mathcal{I}$.

DEFINITION 1.9. *Let $\mathcal{I}$ be a Σ-structure and a an assignment for the individual variables in $\Delta_{\mathcal{I}}$ the interpretation of a term t in $\mathcal{I}$ under the assignment a, denoted by $t^{\mathcal{I}}[a]$ is defined as follows:*

- $c^{\mathcal{I}}[a] = c^{\mathcal{I}}$ *for every constant symbol c;*
- $x^{\mathcal{I}}[a] = a(x)$ *for every individual variable symbol x;*
- $f(t_1, \ldots, t_n)[a] = f^{\mathcal{I}}(t_1^{\mathcal{I}}[a], \ldots, t_n^{\mathcal{I}}[a])$ *for every n-ary functional symbol n and n terms $t_1, \ldots, t_n$.*

Notice that the interpretation of a term, i.e., $t^{\mathcal{I}}[a]$ is "context free", i.e., the interpretation of a term t is independent from the context where it occurs.

REMARK 1. *The interpretation of a term w.r.t. an assignment a depends only on the value that a assigns to the variables that occurs in t. More formally, if a and a' agree on the assignment to the individual variables occurring in t (and might disagree on the assignment to other variables), then $t^{\mathcal{I}}[a] = t^{\mathcal{I}}[a']$.*

The next step is to provide a definition of when a formula is true or false in a Σ-structure $\mathcal{I}$ with respect to the assignment a. We start by defining wen an atomic formula $P(t_1, \ldots, t_n)$ is true in $\mathcal{I}$ w.r.t., the assignment a, in symbols:

$$\mathcal{I} \models P(t_1, \ldots, t_n)[a]$$

DEFINITION 1.10. *An interprertation $\mathcal{I}$ satisfies (makes true) the atomic formula $P(t_1, \ldots, t_n)$ w.r.t. the assignment a, in symbols $\mathcal{I} \models P(t_1, \ldots, t_n)[a]$, if the n-tuple of elements of $\Delta_{\mathcal{I}}$ obtained by interpreting each t_i belongs to $P^{\mathcal{I}}$. In symbols:*

$$\mathcal{I} \models P(t_1, \ldots, t_n)[a] \quad \textit{if} \quad (t_1^{\mathcal{I}}[a], \ldots, t_n^{\mathcal{I}}[a]) \in P^{\mathcal{I}}$$

When Σ contains the equality binary predicate $=$, then it's interpretation is always the same and fixed to the set of pairs (d, d) with $d \in \Delta_{\mathcal{I}}$. As a consequence we have that $\mathcal{I} \models t_1 = t_2$ if and only if $t_1^{\mathcal{I}}[a]$ is equal to $t_2^{\mathcal{I}}[a]$.

In the next definition we use the notation $a_{x \mapsto d}$ for an assignment a an individual variable x and a domain element d, to denote the assignment obtained by modifying a so that $a(x) = d$.

DEFINITION 1.11 (Satisfiability of a formula w.r.t. an assignment). *An interpretation $\mathcal{I}$ satisfies a complex formula ϕ w.r.t. the assignment a according to the following rules:*

$$\begin{array}{rll}
\mathcal{I} \models \phi \wedge \psi[a] & \textit{iff} & \mathcal{I} \models \phi[a] \textit{ and } \mathcal{I} \models \psi[a] \\
\mathcal{I} \models \phi \vee \psi[a] & \textit{iff} & \mathcal{I} \models \phi[a] \textit{ or } \mathcal{I} \models \psi[a] \\
\mathcal{I} \models \phi \rightarrow \psi[a] & \textit{iff} & \mathcal{I} \not\models \phi[a] \textit{ or } \mathcal{I} \models \psi[a] \\
\mathcal{I} \models \neg\phi[a] & \textit{iff} & \mathcal{I} \not\models \phi[a] \\
\mathcal{I} \models \phi \equiv \psi[a] & \textit{iff} & \mathcal{I} \models \phi[a] \textit{ iff } \mathcal{I} \models \psi[a] \\
\mathcal{I} \models \exists x\phi[a] & \textit{iff} & \textit{there is a } d \in \Delta_{\mathcal{I}} \textit{ such that } \mathcal{I} \models \phi[a_{x \leftarrow d}] \\
\mathcal{I} \models \forall x\phi[a] & \textit{iff} & \textit{for all } d \in \Delta_{\mathcal{I}}, \mathcal{I} \models \phi[a_{x \leftarrow d}]
\end{array}$$

When ϕ is a closed formula then $\mathcal{I} \models \phi[a]$ iff $\mathcal{I} \models \phi[a']$ for any assignment a', therefore the assignment does not play any role and we simplify the notaiton with $\mathcal{I} \models \phi$ skipping the assignment.

The notation $\phi[a]$ is very similar to the notation $\phi[\sigma]$ used for substitution where σ is a substitution of variables with terms. Though they are related, they should not be confused. a and σ are different concepts, since a maps variables into elmeents of the domain $\Delta_{\mathcal{I}}$, while σ maps variables in terms in the signature Σ. While $\phi[\sigma]$ is a formula, obtained by replacing each free occurrence of a variable x with $\sigma(x)$, $\phi[a]$ is just a formula ϕ and an assignmennt a, no transformation is einvolved in $\phi[a]$. One should not think that $\phi[a]$ is a formula obtained by replacing any variable x with $a(x) \in \Delta^{\mathcal{I}}$. Indeed the result of this replacement is not a formula since $\alpha[x)$ is en elment of the domian and not of the signature.

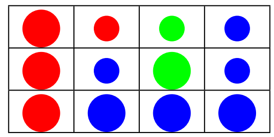
However there is a connection between the two expressions. This is stated by the followin proposition

PROPOSITION 1.2. *If $t^{\mathcal{I}}[a] = d$*

(1) for every term s, $s(x)^{\mathcal{I}}[a_{x \leftarrow d}] = s(t)^{\mathcal{I}}[a]$.
(2) then $\mathcal{I} \models \phi(t)$ iff $\mathcal{I} \models \phi(x)[a_{x \leftarrow d}]$.

PROOF. The proof is by induction on the terms and the formulas. □

EXAMPLE 1.15. *Consider*[2] *structures represented by a rectangular grid of "dots" Each dot has a color (red, blue, or green) and a size (small or large). An example of such a structure is shown in the following picture*



The logical language we use to describe our dot world has unary predicates red *,* green*,* blue*,* small *and* large*, which are interpreted in the obvious ways. The binary predicate* adj *is interpreted in the adiacency relation that contains the pairs of dots* (d, d') *contained in two slots with a wall in common. The binary predicates* sameColor*,* sameSize*,* sameRow*, and* sameColumn *are interpreted in the relation that contains the pairs of dots* (d, d') *that have the same color, the same size, they are on the same row, or on the same column respectively. Finally, the binary predicate* leftOf *is interpreted in the set of pair of dots* (d, d') *sich that* d *is left of* d' *regardless of what rows the dots are in. The interpretations of* rightOf*, above, and* below *are similar.*

Consider the following sentences:

(1) $\forall x(\mathsf{green}(x) \vee \mathsf{blue}(x))$
(2) $\exists x, y(\mathsf{adj}(x, y) \wedge \mathsf{green}(x) \wedge \mathsf{green}(y))$
(3) $\exists x((\exists z \mathsf{rightOf}(z, x)) \wedge (\forall y(\mathsf{leftOf}(x, y) \to \mathsf{blue}(y) \vee \mathsf{small}(y))))$
(4) $\forall x(\mathsf{large}(x) \to \exists y(\mathsf{small}(y) \wedge \mathsf{adj}(x, y)))$
(5) $\forall x(\mathsf{green}(x) \to \exists y(\mathsf{sameRow}(x, y) \wedge \mathsf{blue}(y)))$
(6) $\forall x, y(\mathsf{sameRow}(x, y) \wedge \mathsf{sameColumn}(x, y) \to x = y)$
(7) $\exists x \forall y(\mathsf{adj}(x, y) \to \neg \mathsf{sameSize}(x, y))$
(8) $\forall x \exists y(\mathsf{adj}(x, y) \wedge \mathsf{sameColor}(x, y))$
(9) $\exists y \forall x(\mathsf{adj}(x, y) \wedge \mathsf{sameColor}(x, y))$
(10) $\exists x(\mathsf{blue}(x) \wedge \exists y(\mathsf{green}(y) \wedge \mathsf{above}(x, y)))$

We can evaluate them in the model shown above: There they have the following truth values:

(1) *false*
(2) *true*
(3) *true*
(4) *false*
(5) *true*
(6) *true*
(7) *false*
(8) *true*
(9) *false*
(10) *true*

For each sentence, see if you can find a model that makes the sentence true, and another that makes it false. For an extra challenge, try to make all of the sentences true simultaneously. Notice that you can use any number of rows and any number of columns.

[2] `https://leanprover.github.io/logic_and_proof/semantics_of_first_order_logic.html`

REMARK 2. *Notice that, t does not contain any variable then in evaluating $t^{\mathcal{I}}[a]$ the assignmnet a does not play any role. Indeet it is easy to see that $t^{\mathcal{I}}[a] = t^{\mathcal{I}}[a']$ for every pair of assignments a and a'. We call such a term,* ground *term, and since we don't need the assignment to variables to evaluate it, we simplify $t^{\mathcal{I}}[a]$ with $t^{\mathcal{I}}$.*

REMARK 3. *Consider now a term t that contains the variables x, (we denote this with $t(x)$) and let a and a' be two assignments such that $a(x) = a'(x)$. One can see from the inductive definition that $t^{\mathcal{I}}[a] = t^{\mathcal{I}}[a']$, since the only variable that occours in t is x. This can be extended to terms that contains n variables and to formulas that contains n* free variables*.*

PROPOSITION 1.3.

(1) If $t(x_1, \ldots, x_n)$ is a term that contains the variables $x_1, \ldots, x_n$, then if $a(x_i) = a'(x_i)$ for every $1 \leq i \leq n$, then $t^{\mathcal{I}}[a] = t^{\mathcal{I}}[a']$.
(2) If $\phi(x_1, \ldots, x_n)$ is a formiula that contains the free variables $x_1, \ldots, x_n$, then if $a(x_i) = a'(x_i)$ for every $1 \leq i \leq n$, then $\mathcal{I} \models \phi(x_1, \ldots, x_n)[a]$ iff $\mathcal{I} \models \phi(x_1, \ldots, x_n)[a']$.

3.1. Satisfiability, Validity, and Logical Consequence. In every logical language the notions of satisfiable formula, valid formula, unsatisfiable fomrula, and the logical consequence relation is defined on the basis of the semantics of the logic. The semantics of a logic defines how the symbols of the logic can be interpreted in some structure called interpretation, and defines the satisfaction relation, usually denoted by $\models$ between interpretation and the formulas of the logical language. The definition of the above concepts is almost the same in all the logics. In the following we report the specific definition for the first order logic, but the student is invited to compare these definitions with the analogous definitions given for propositional logic, in order to recognize the analogies and the (small) differences.

DEFINITION 1.12 (Model, satisfiability and validity).

(1) An interpretation $\mathcal{I}$ of a signature Σ is a model of a first order formula ϕ in the signature Σ w.r.t. the assignment a, if $\phi[a]$ is evaluated true in $\mathcal{I}$, in symbols if

$$\mathcal{I} \models \phi[a]$$

(2) A formula ϕ is satisfiable *if there is some interpretation $\mathcal{I}$ and some assignment a such that $\mathcal{I} \models \phi[a]$.*
(3) A formula ϕ is unsatisfiable *if it is not satisfiable.*
(4) A formula ϕ is valid *if every $\mathcal{I}$ and every assignment a $\mathcal{I} \models \phi[a]$*

REMARK 4. *Consider $\phi(x_1, \ldots, x_n)$ to be any first-order formula with free variables $x_1, \ldots, x_n$. We say that*

- *the sentence $\exists x_1, \ldots, \exists x_n.\phi(x_1, \ldots, x_n)$ is the existential closure of ϕ;*
- *the sentence $\forall x_1, \ldots, \forall x_n.\phi(x_1, \ldots, x_n)$ is the universal closure of ϕ.*

Satisfiability and validify of open formulas (i.e., formuilas with free variables) cen be reduced to satisfiability and validity of sentences (closed formulas = formulas without free variabels). This is stated in the following proposition.

PROPOSITION 1.4.

(1) $\phi(x_1, \ldots, x_n)$ is satisfiable iff $\exists x_1 \ldots x_n.\phi(x_1, \ldots, x_n)$ is satisfiable ;
(2) $\phi(x_1, \ldots, x_n)$ is valid iff $\forall x_1 \ldots x_n.\phi(x_1, \ldots, x_n)$ is valid.

Proof. By exercize □

The relation of logical consequence, expresses the fact that one formula is true under the hypothesis that a set of formulas are true.

Definition 1.13 (Logical Consequence). *A formula ϕ is a logical consequence of a set of formulas Γ, in symbols $\Gamma \models \phi$, if for all interpretations $\mathcal{I}$ and for all assignment a*

$$\mathcal{I} \models \Gamma[a] \qquad \Longrightarrow \qquad \mathcal{I} \models \phi[a]$$

where $\mathcal{I} \models \Gamma[a]$ means that $\mathcal{I}$ satisfies all the formulas in Γ under a.

3.2. Properties of quantifiers. The most important extension of FOL w.r.t., propositional logic are quantifiers. In the following let us see the most important properties of universal and existential quantifiers.

We start by showing that the formula $\forall x \phi(x)$, read as "the propoerty specified by the formula $\phi(x)$ is true for every x", implies $\phi(t)$, i.e., that ϕ holds for every closed term t.

Proposition 1.5. *For every term t and formula $\phi(x)$ the formula*

$$\forall x \phi(x) \to \phi(t)$$

is valid.

Proof. We have to prove that $\mathcal{I} \models \forall x \phi(x) \to \phi(t)[a]$ for every interpretation $\mathcal{I}$ and any assignment a. This is equivalent to prove that if $\mathcal{I} \models \forall x \phi(x)[a]$ then $\mathcal{I} \models \phi(t)[a]$.

$$\begin{aligned}
\mathcal{I} \models \forall x \phi(x)[a] &\Longleftrightarrow \mathcal{I} \models \phi(x)[a_{x \leftarrow d}] \text{ for all } d \in \Delta^{\mathcal{I}} \\
&\Longrightarrow \mathcal{I} \models \phi(x)[a_{x \leftarrow d}] \text{ for } d = t^{\mathcal{I}}[a] \\
&\Longrightarrow \mathcal{I} \models \phi(t)[a] \text{ by Proposition 1.2}
\end{aligned}$$

□

The opposite properrty holds for the existential quantifier

Proposition 1.6. *For every term t and formula $\phi(x)$ the formula*

$$\phi(t) \to \exists x \phi(x)$$

n is valid.

A second important property is the duality of the two quantifiers, and the fact that they are definable one in terms of the other. This property is shown by the fact that for every formula $\phi(x)$ with one free variable x, we have that the following formulas are valid.

$$\forall x.\phi(x) \leftrightarrow \neg \exists x \neg \phi(x)$$
$$\exists x.\phi(x) \leftrightarrow \neg \forall x \neg \phi(x)$$

A third property concern the vacuous quantification. If a variable x is not free in ϕ then quantifying ϕ on x does not have any effect. Indeed the formulas

$$\forall x \phi \leftrightarrow \phi$$
$$\exists x \phi \leftrightarrow \phi$$

are valid when x is not free in ϕ.

Let us now see how quantifiers interacts with connectives. We start by noticing that $\forall x$ commutes with $\wedge$. Indeed thhe formula

$$\forall x(\phi \wedge \psi) \leftrightarrow \forall x\phi \wedge \forall x\psi$$

is valid. Differently we have that $\forall$ does not commute with $\vee$. Indeed the formula

$$\forall x(\phi \vee \psi) \leftrightarrow \forall x\phi \vee \forall x\psi$$

is not valid. Consider the following example.

EXAMPLE 1.16. *Let P be a unary predicate, we have that $\forall x(P(x) \vee \neg P(x))$ is true in every interpretation $\mathcal{I}$, as every of the domain $\Delta^{\mathcal{I}}$ is either in $P^{\mathcal{I}}$ or not in $P^{\mathcal{I}}$; while $\forall x P(x) \vee \forall x \neg P(x)$ is true only in the interpretation where P is interpreted either in the empty set of in the entire domain $\Delta^{\mathcal{I}}$.*

However, we have that this formula holds in one direction. indeed the formula

$$\forall x\phi \vee \forall x\psi \leftrightarrow \forall x(\phi \vee \psi)$$

is valid.

Let us now consider $\forall$ and $\neg$. In English saying "not all flowers are beautifull" it is not the same as saying "all flowers are not beautifull". This holds also in FOL. Inded the formula ¡

$$\forall x\neg\phi \leftrightarrow \neg\forall x\phi$$

is not valid.

The relationship between $\forall$ and $\rightarrow$. We have that the formula

$$\forall x\phi \rightarrow \forall x\psi \leftrightarrow \forall x(\phi \rightarrow \psi)$$

is not valid. Consider for instan the following interpretation.

EXAMPLE 1.17. *$\forall x\neg P(x) \rightarrow \forall x P(x)$ is satisfied by all the interpretations. where P is not interpreteed in the empty set. while $\forall x(\neg P(x) \rightarrow P(x))$ is satisfied only in the interpretations where P is interpreted in the emtore domain (since it is equivalent to $\forall x P(x)$).*

As happens for the disjunction we have that one direction is valid. Indeed

$$\forall x(\phi \rightarrow \psi) \rightarrow \forall x\phi \rightarrow \forall x\psi$$

is valid.

As a final property, we show that $\forall$ commutes with $\vee$ and $\rightarrow$ under some conditions. IN particular we have that if x is not free in ϕ we have that

$$\forall x(\phi \vee \psi) \leftrightarrow \phi \vee \forall x\psi$$
$$\forall x(\phi \rightarrow \psi) \leftrightarrow \phi \rightarrow \forall x\psi$$

are both valid.

Let us perform the same analisys for the existential quantifier. We summarize the results in the following table

$\exists x(\phi \wedge \psi) \leftrightarrow \exists x\phi \wedge \exists x\psi$	is not valid
$\exists x(\phi \wedge \psi) \rightarrow \exists x\phi \wedge \exists x\psi$	is valid
$\exists x(\phi \vee \psi) \leftrightarrow \exists x\phi \vee \exists x\psi$	is valid
$\exists x\neg\phi \leftrightarrow \neg\exists x\phi$	is not valid
$\neg\exists x\phi \rightarrow \exists x\neg\phi$	is valid
$\exists x(\phi \rightarrow \psi) \leftrightarrow \exists x\phi \rightarrow \exists x\psi$	is not valid
$(\exists x\phi \rightarrow \exists x\psi) \rightarrow \exists x(\phi \rightarrow \psi)$	is valid
$\exists x(\phi \wedge \psi) \leftrightarrow \phi \wedge \exists x\psi$	is valid if x is not free in ϕ
$\exists x(\phi \rightarrow \psi) \leftrightarrow \phi \rightarrow \exists x\psi$	is valid if x is not free in ϕ

Let us now see how quantifiers interacts one another. The formula

$$\forall x\phi \rightarrow \exists x\phi$$

is valid. The validity is guaranteed by the fact that the domain $\Delta^{\mathcal{I}}$ of any first order interpretation $\mathcal{I}$ is not empty. Indeed if $\mathcal{I} \models \forall x\phi$, ϕ is true for all the elements of the domain $\Delta^{\mathcal{I}}$, and since $\Delta^{\mathcal{I}}$ is not empty then there exists at least one element of the domain for which the property $\phi(x)$ is true. Obviously the converse

$$\exists x\phi \rightarrow \forall x\phi$$

Quantifiers of the same type can permute. Therefore

$$\forall x\forall y\phi \leftrightarrow \forall y\forall x\phi$$
$$\exists x\exists y\phi \leftrightarrow \exists y\exists x\phi$$

are valid formulas. Instead, quantifiers of different type do not permute Indeed

$$\forall x\exists y\phi \leftrightarrow \exists y\forall x\phi$$

is not valid. An intuitive counteressample of the previous equivalence can be obtained by a binary relation R. We have that $\forall x\exists yR(x, y)$ means that "exery x is related via R with some y, while $\exists y\forall xR(x, y)$ means that there is a y with which every x is related. A concrete example

$$\forall x\exists y\, \mathsf{supervisor}(y, x)$$

means that everybody has a supervisor. while the formula

$$\exists y\forall x\, \mathsf{supervisor}(y, x)$$

everybody has the same supervisor, or there is somebody who is supervising everybody.

4. Exercises

Exercise 1:

Draw the parse tree of the formula $\forall x\, P(x) \rightarrow \exists y\exists z\, Q(y, z) \wedge \neg\exists s\, R(x)$ respecting the precedence of operators.

Exercise 2:

For each the following formulas say which are the free and the bound occurrences of variables in the following formula. For the bound variables indicate the quantifier that bounds it.

(1) $\exists x(E(x,y) \wedge \exists y \neg E(y,x))$;
(2) $\forall x(\forall z A(x,z) \rightarrow \exists y(R(x,y) \wedge \forall x P(x,z)))$;
(3) $\forall z(A(x) \wedge B(x,y)) \rightarrow \forall z A(z,z)$.

Exercise 3:

Suppose that Σ contains the following symbols with the associated intuitive meaning:

T	Trento	
R	Rome	
I	Iitaly	
M	Marocco	
L	Lorenzo	
F	Francesca	
$major(x)$	The major of $x motherOf(x)$	the mother of x
$fatherOf(x)$	the father of x	
$nationality(x)$	The nationality (country) of x	
$homeTown(x)$	The hometown of x	
$capital(x)$	the capital of x	
$route(x,y)$	the route from x to y	
$routeThroug(x,y,z)$	The route from x to y passing through z	

write the terms that corresponds to the following English description

(1) the capital of Italy;
(2) the hometown of Lorenzo;
(3) the route that connects the hometowns of Lorenzo and Francesca;
(4) the capital of the nationality of Francesca;
(5) the route from the hometown of the father of Francesca and the mother of Lorenzo passing through the capital of Marocco;
(6) the route going from the hometown of the major of the capital of Marocco to the hometown of his/her mother qspassing through the hometown of his/her grandfather.

Exercise 4:

Using the following symbols with the associated intuitive meaning

a	Adam
e	Eva
c	Cid
$\mathsf{B}(x)$	x is blond
$\mathsf{C}(x)$	x is a cat
$\mathsf{L}(x, y)$	x loves y
$\mathsf{T}(x, y)$	x is taller than y

Transcribe the following FOL sentences into English:

(1) $\mathsf{T(c,e)}$
(2) $\mathsf{L(c,e)}$
(3) $\neg\mathsf{T(c,c)}$
(4) $\mathsf{B(c)}$
(5) $\mathsf{T(c,e)} \to \mathsf{L(c,e)}$
(6) $\mathsf{L(c,e)} \vee \mathsf{L(c,c)}$
(7) $\neg(\mathsf{L(c,e)} \wedge \mathsf{L(c,a)})$
(8) $\mathsf{B(c)} \leftrightarrow (\mathsf{L(c,e)} \vee \mathsf{L(c,c)})$

Transcribe the following English sentences into sentences first order logic;

(1) Cid is a cat.
(2) Cid is taller than Adam.
(3) Either Cid is a cat or he is taller than Adam.
(4) If Cid is taller than Eve then he loves her.
(5) Cid loves Eve if he is taller than she is.
(6) Eve loves both Adam and Cid.
(7) Eve loves either Adam or Cid.
(8) Either Adam loves Eve or Eve loves Adam, but both love Cid.
(9) Only if Cid is a cat does Eve love him.
(10) Eve is taller than but does not love Cid.

Exercise 5:

Transcribe the following english sentence in FOL

(1) Everyone loves Eve.
(2) Eve loves somebody.
(3) Eve loves everyone.
(4) Some cat loves some dog.
(5) Somebody is neither a cat nor a dog.
(6) Someone blond loves Eve.
(7) Some cat is blond.
(8) Somebody loves all cats.
(9) No cat is a dog.
(10) Someone loves someone.
(11) Everybody loves everyone.
(12) Someone loves everyone.
(13) Someone is loved by everyone.
(14) Everyone loves someone.

(15) Everyone is loved by somebody.

Exercise 6:

For each of the following formulas indicate: (a) the scope of the quantifiers (b) the free variables, and (c) whether it is a sentence (closed formula)

(1) $\forall x(A(x,y) \wedge B(x))$
(2) $\forall x(\forall y(A(x,y) \rightarrow B(x))$
(3) $\forall x(\forall y(A(x,y))) \rightarrow B(x)$
(4) $\exists x(: \forall y(A(x,y)))$
(5) $\forall x(A(x,y)) \wedge B(x)$
(6) $\forall x(A(x,x)) \wedge \forall y(B(y))$

Solution

□

Exercise 7:

Say if the term $f(x,y)$ is free for z in the following formulas:

(1) $\forall z(P(x,y,z) \rightarrow \exists y Q(x,y))$
(2) $\forall x(R(y,z) \wedge existszQ(z,y))$
(3) $P(x,y,z) \wedge \exists x Q(x,z)$

Exercise 8:

Apply the substitution $x/a, y/b, z/c$ to the formula $\forall z P(x,y,z) \rightarrow \forall y(Q(x,y) \wedge \exists x R(x,z))$

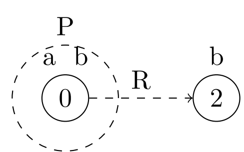
Exercise 9:

Find an interpretation that satisfies the formula

$$P(a) \wedge P(b) \wedge \forall x(P(x) \rightarrow \exists y.(R(x,y) \wedge \neg P(y)))$$

Solution

- $\Delta^{\mathcal{I}} = \{0,1\}$
- $a^{\mathcal{I}} = 0$
- $b^{\mathcal{I}} = 0$
- $P^{|} = \{0\}$
- $R^{\mathcal{I}} = \{(0,1)\}$



□

Exercise 10:

Translate the following sentences in FOL.

(1) Everything is bitter or sweet
(2) Either everything is bitter or everything is sweet

(3) There is somebody who is loved by everyone
(4) Nobody is loved by no one
(5) If someone is noisy, everybody is annoyed
(6) Frogs are green.
(7) Frogs are not green.
(8) No frog is green.
(9) Some frogs are not green.
(10) A mechanic likes Bob.
(11) A mechanic likes herself.
(12) Every mechanic likes Bob.
(13) Some mechanic likes every nurse.
(14) There is a mechanic who is liked by every nurse.

Solution

(1) Everything is bitter or sweet

$$\forall x(\mathsf{bitter}(x) \lor \mathsf{sweet}(x))$$

(2) Either everything is bitter or everything is sweet

$$\forall x\ \mathsf{bitter}(x) \lor \forall x\ \mathsf{sweet}(x)$$

(3) There is somebody who is loved by everyone

$$\exists x \forall y\ \mathsf{loves}(y, x)$$

(4) Nobody is loved by no one

$$\neg\exists x \neg\exists y\ \mathsf{loves}(y, x)$$

(5) If someone is noisy, everybody is annoyed

$$\exists x\ \mathsf{noisy}(x) \to \forall y\ \mathsf{annoyed}(y)$$

(6) Frogs are green.

$$\forall x\ (\mathsf{frog}(x) \to \mathsf{green}(x))$$

(7) Frogs are not green.

$$\forall x\ (\mathsf{frog}(x) \to \neg\mathsf{green}(x))$$

(8) No frog is green.

$$\neg\exists x\ (\mathsf{frog}(x) \land \mathsf{green}(x))$$

(9) Some frogs are not green.

$$\exists x\ (\mathsf{frog}(x) \land \neg\mathsf{green}(x))$$

(10) A mechanic likes Bob

$$\exists x(\mathsf{mechanic}(x) \land \mathsf{likes}(x, \mathsf{Bob})$$

(11) A mechanic likes herself.

$$\exists x(\mathsf{mechanic}(x) \land \mathsf{likes}(x, x))$$

(12) Every mechanic likes Bob.

$$\forall x(\mathsf{mechanic}(x) \to \mathsf{likes}(x, \mathsf{Bob}))$$

(13) Some mechanic likes every nurse.

$$\exists x(\mathsf{mechanic}(x) \wedge \forall y(\mathsf{nurse}(y) \rightarrow \mathsf{likes}(x, y)))$$

(14) There is a mechanic who is liked by every nurse.

$$\exists x(\mathsf{mechanic}(x) \wedge \forall y(\mathsf{nurse}(y) \rightarrow \mathsf{likes}(y, x)))$$

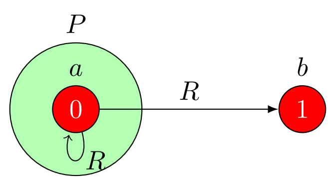
□

Exercise 11:

Given the FOL interpretation $\mathcal{I}$ defined on the domain $\Delta_{\mathcal{I}} = \{0, 1\}$ and the interpretation: $\mathcal{I}(P) = \{0\}$ and $\mathcal{I}(R) = \{(0, 0), (0, 1)\}$ $\mathcal{I}(a) = 0$ and $\mathcal{I}(b) = 1$. Verify whether the following formulas are true in M:

(1) $\forall x P(x)$
(2) $P(a)$
(3) $P(b)$
(4) $R(a, b)$
(5) $\neg R(a, a)$
(6) $\exists x R(a, x)$
(7) $\forall x R(x, b)$
(8) $\forall x R(x, x) \rightarrow P(x)$
(9) $\forall x \neg R(a, x) \rightarrow P(x)$
(10) $\forall x(P(x) \rightarrow \neg R(a, x))$

Solution The interpretation $\mathcal{I}$ can be graphically represented as follows:



Let us now check the truth value of the formulas

(1) $\forall x P(x)$ is false since thate is an element of the domain which is not in $\mathcal{I}(P)$;
(2) $P(a)$ is true since $\mathcal{I}(a) = 0 \in \mathcal{I}(P)$;
(3) $P(b)$ is false since $\mathcal{I}(b) = 1 \notin \mathcal{I}(P)$;
(4) $R(a, b)$ is true since $(\mathcal{I}(a), \mathcal{I}(b)) = (0, 1) \in \mathcal{I}(R)$;
(5) $\neg R(a, a)$ is false since $(\mathcal{I}(a), \mathcal{I}(a)) = (0, 0) \in \mathcal{I}(R)$, which makes $P(a, a)$ true and therefore $\neg P(a, a)$ false;
(6) $\exists x R(a, x)$ is true since if x is assigned to 0 we have that the pair $(\mathcal{I}(a), x) = (0, 0) \in \mathcal{I}(R)$;
(7) $\forall x R(x, b)$ is false because, if we assigne 0 to x we have that the pair $(x, \mathcal{I}(b)) = (0, 1) \notin \mathcal{I}(R)$;
(8) $\forall x R(x, x) \rightarrow P(x)$ is true. To show this we have to chrck the truth of $R(x, x) \rightarrow P(x)$ for all the possible assignments of x. If x is assigned to 0, $\mathcal{I} \models P(x)[x := 0]$ and therefore $\mathcal{I} \models R(x, x) \rightarrow P(x)[x := 0]$; If x is assigned to 1, then we have that $\mathcal{I} \not\models R(x, x)[x := 1]$, and therefore $\mathcal{I} \models R(x, x) \rightarrow P(x)[x := 1]$, Since $R(x, x) \rightarrow P(x)$ is true for all the assignments of x, we can conclude that $\mathcal{I} \models \forall x(R(x, x) \rightarrow P(x))$ is true..

(9) $\forall x \neg R(a,x) \to P(x)$ is true. As in the previous case we have to check for all the assignments of x. Since we have that $\mathcal{I} \models R(a,x)[x := 0]$ we have that $\mathcal{I} \models \neg R(a,x) \to P(x)[x := 0]$; We also have that $\mathcal{I} \models R(a,x)[x := 1]$, and therefore $\mathcal{I} \models \neg R(a,x) \to P(x)[x := 1]$. We can therefore conclude that $\mathcal{I} \models \forall x(\neg R(a,x) \to P(x)$

(10) $\forall x(P(x) \to \neg R(a,x))$ is false since if x is assignmet to 0 we have that $\mathcal{I} \not\models P(x) \to \neg R(a,x)[x = 0]$ as $0 \in \mathcal{I}(P)$ and $(0,0) \in \mathcal{I}(R)$.

□

Exercise 12:
Consider the formula

$$\forall x \forall y (P(x,y) \wedge Q(y,x))$$

Give an interpretation that satisfies it in the domain $D = \{\text{John}, \text{Paul}, \text{Mary}\}$ **Exercise 13:**

Show the following equivalence:

$$(\forall x P(x) \wedge \forall x Q(x)) \leftrightarrow \forall x(P(x) \wedge Q(x)) \tag{3}$$

is valid

Solution Let us consider an interpretation $\mathcal{I}$ on the domain Δ, To prove that (3) is valid, we show that $\mathcal{I} \models \forall x P(x) \wedge \forall x Q(x)$ if and only if $\mathcal{I} \models \forall x(P(x) \wedge Q(x))$.

$$\begin{aligned}
\mathcal{I} \models \forall x(P(x) \wedge Q(x) &\Leftrightarrow \text{for all } d \in \Delta, \mathcal{I} \models P(x) \wedge Q(x)[a_{x \leftarrow d}] \\
&\Leftrightarrow \text{for all } d \in \Delta, \mathcal{I} \models P(x) \text{ and } \mathcal{I} \models Q(x)[a_{x \leftarrow d}] \\
&\Leftrightarrow \mathcal{I} \models \forall x P(x) \text{ and } \mathcal{I} \models \forall x Q(x) \\
&\Leftrightarrow \mathcal{I} \models \forall x P(x) \wedge \forall x Q(x)
\end{aligned}$$

□

Exercise 14:
Decide if

$$\forall x Q(x) \to \forall x P(x) \leftrightarrow \exists z \forall y (Q(y) \vee P(z))$$

is valid, satisfiable, non valid, or unsatisfiable.

Exercise 15:
Decide if

$$\forall x Q(x) \to \forall x P(x) \leftrightarrow \exists z \forall y (Q(y) \vee P(z))$$

is valid, satisfiable, non valid, or unsatisfiable, and explain your answer.

Solution One can try to rewrite the left part of the formula in order to obtain a formula equivalent to the right part.

$$\begin{gathered}
\forall x Q(x) \to \forall x P(x) \\
\neg \forall x Q(x) \vee \forall x P(x) \\
\exists x \neg Q(x) \vee \forall x P(x) \\
\exists z \neg Q(z) \vee \forall y P(y) \\
\exists z \forall y (\neg Q(z) \vee P(y))
\end{gathered}$$

One can now easily see that the obtained formula cannot be equivalent to $\exists z\forall y(\neg Q(z)\vee P(y))$. For instance the interpretation

$$\mathcal{I}(Q) = \{a, b\} \qquad\qquad \mathcal{I}(P) = \{a\}$$

satisfies $\exists x\forall y(Q(x) \vee P(y))$. Indeed $\mathcal{I} \models Q(a)$ and therefore $\mathcal{I} \models \forall y(Q(a) \vee P(y))$ and therefore $\mathcal{I} \models \exists x\forall y(Q(x) \wedge P(y))$. On the other hand $\mathcal{I} \not\models \exists x\forall y(\neg Q(x) \vee P(y))$ since there are no constants for which $Q(x)$ is true, and $Q(x)$ is not true for all the elements of the domain. Since we have foound an interpretation that satisfies the right hand side and do not satisfy the left hand side of the equivalence, the equivalence is not satisfied by an interpretation and therefore it is not valid.

The formula is satisfiable, since the interpretation that makes $\forall xP(x)$ always true, satisfies noth the left and the right hand side of the equivalence.

□

Exercise 16:

Show that the following formula is not valid

$$(\forall xP(x) \vee \forall xQ(x)) \leftrightarrow \forall x(P(x) \vee Q(x))$$

and provide an english sentence by instantiating the predicates P and Q in english adjectives with is intuitively false.

Solution Consider the first order interpretation $\mathcal{I}$ on the domain of natural numbers $\mathbb{N} = \{1, 2, 3, \dots\}$, where P is interpreted in the set of the even numbers, and Q in the set of odd numbers. We have that $\mathcal{I} \models \forall x(P(x)\vee Q(x)$ as every natural number is either odd or even. However $\mathcal{I} \not\models \forall xP(x)$ and $\mathcal{I} \not\models \forall xQ(x)$ since not all the numbers are even, or all the numbers are odd, and therefore $\mathcal{I} \not\models \forall xP(x) \vee \forall xQ(x)$.

□

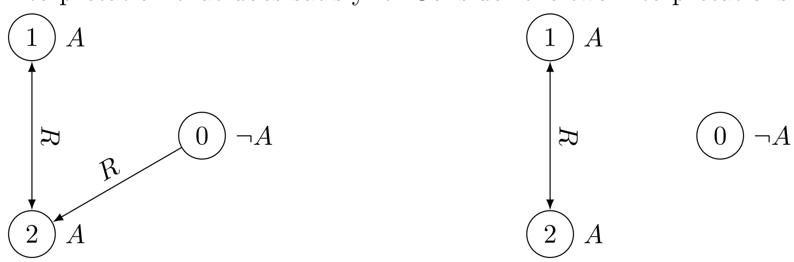
Exercise 17:

For the following formula check if it is (a) valid, (b) satisfiable, (c) not valid, (d) unsatisfiable. Notice that more than one option is possible.

$$\forall x\exists y(A(x) \leftrightarrow R(x, y)) \leftrightarrow \forall x(A(x) \leftrightarrow \exists yR(x, y))$$

For each choice provide an argument that support your choice.

Solution The formula is not valid and satisfiable. Indeed we can provide a counterexample, i.e., an interpretation that does not satisfy the formula, and an example i.e, an interpretation that does satisfy it. Consider the two interpretations



The leftmost interpretation does not satisfy the formula. Indeed $\exists y(A(x) \leftrightarrow R(x, y))$ is true for all the assignments to x, but $(A(x) \leftrightarrow \exists yR(x, y))$ is false when x is assigned to 0. Indeed, while $\exists yR(x, y)[x \leftarrow 0]$ is true, because $R(x, y)[x \leftarrow 0, y \leftarrow 2]$ is true, $A(x)[x \leftarrow 0$ is false, and therefore the equivalence $A(x) \leftrightarrow \exists yR(x, y)$ is false when x assigned to 0.

Instead, one can easily see that, the rightmost interpretation instead satisfies the formula. □

Exercise 18:

Find a formula ϕ which is true for some interpretation with infinite domain, but false for all interpretation with finite domains.

Solution Define ϕ to be the conjunciton of:

$$\forall xyz(p(x,y) \land p(y,z) \rightarrow p(x,z))$$
$$\forall x \neg p(x,x)$$
$$\forall x \exists y p(x,y)$$

We can verify that, for any finite interpretation, ϕ is false. On the other haqnd, ϕ is true under the interpretation $\mathcal{I}$ with $\Delta^{\mathcal{I}} = \mathbb{N}$ (the set of natural numbers and $\mathcal{I}(R) = \{(n,m) \in \mathbb{N}^2 \mid n < m\}$ □