CHAPTER 2. FIRST ORDER LOGIC

1. Introduction

First order logic is a much richer system than sentential logic. Its interpretations include the usual structures of mathematics, and its sentences enable us to express many properties of these structures. For example, consider the following English sentence:

Everything greater than 0 *has a square root.*

This can be interpreted, for example, in arithmetic on $\mathbb{Q}$, the set of rational numbers, where it is false, and in arithmetic on $\mathbb{R}$, the set of real numbers, where it is true. How could we express this in a formal language? The following is at least a first approximation.

For all x $(0 < x \rightarrow$ *there is some* y *such that* $x = y \cdot y)$.

A formal language that could express this would need symbols for

- quantifiers *for all* and *there exists*,
- variables, like x and y, which range over the elements of the set we are talking about,
- functions, like $\cdot$,
- relations, like $=$ and $<$,
- constants, like 0, which name fixed elements of the domain,
- sentential connectives,
- parentheses.

We will use the symbols $\forall$ for *for all* and $\exists$ for *there exists*. Thus the above sentence could be written as

$$\forall x\big(0 < x \rightarrow \exists y(x = y \cdot y)\big)$$

We will normally take formalization one step further. We allow interpretations which interpret $<$, for example, as *any* binary relation on the domain, not just something that "looks" like an order. To emphasize this we use neutral symbols, like R, F, and c for the relations, functions, and constants of the language. The only exception is $=$ whose interpretation is fixed. Thus, fully formalized, the sentence we are looking at would be written as

$$\forall x\big(R(c, x) \rightarrow \exists y(x = F(y, y))\big)$$

An interpretation for this would specify what set of objects is referred to (the *domain* of the interpretation), and how to interpret the symbols c, R, F. One such interpretation has domain $\mathbb{Z}$, the set of all integers, and interprets c as 2, $R(x, y)$ as "x divides y", and F as $+$. Note that the sentence is true in this interpretation.

Note that this sentence is built from pieces which are not themselves sentences, for example $\exists y(x = F(y, y))$. In any interpretation this defines a subset of the domain, namely the set of all x in the domain such that $x = F(y, y)$ for some y in the domain. In the interpretation with domain $\mathbb{R}$ with c, R, F interpreted as $0, < \cdot$, this is the set of all non-negative reals. In the interpretation in the preceding paragraph, this is the set of even integers. Such an expression will be called a *formula*, and we will need to know how to interpret formulas before defining when a sentence is true.

2. Functions, Relations, and Structures

Let A be a non-empty set and let $n \in \mathbb{N}$. An *n-ary function f on A* assigns to every n-tuple $a_1, \dots, a_n$ from A a unique element $f(a_1, \dots, a_n) \in A$. Note, for example, that subtraction is not considered a binary function on $\mathbb{N}$ since $3 - 7 \notin \mathbb{N}$.

If F is an n-ary function on A and $G_1, \dots, G_n$ are each k-ary functions on A then we can define a k-ary function H on A by *composition* as $H(a_1, \dots, a_k) = F(G_1(a_1, \dots, a_k), \dots, G_n(a_1, \dots, a_k))$ for all $a_1, \dots, a_k \in A$. For example, from $+$ and $\cdot$ on $\mathbb{N}$ we can define $(x+y)\cdot(x\cdot y)$, $x\cdot y+y\cdot x$, etc. If the functions $G_1, \dots, G_n$ do not all have the same arity or variable list, we add dummy variables until they do. Thus we can also obtain $x \cdot (y + x)$, $x \cdot x + y \cdot y$, etc.

An *n-ary relation R on A* picks out a set of n-tuples of elements of A, those for which the relation *holds*, written $R(a_1, \dots, a_n)$ *holds*. For all other n-tuples the relation *fails*, written $R(a_1, \dots, a_n)$ *fails*. Note that a unary relation picks out a subset of A. One specific relation of importance is the binary relation of equality on A , $=$, which holds between a pair of elements of A precisely when they are the same.

A *structure* consists of some non-empty set A, called the *domain* or *universe* of the structure, together with some collection of functions and relations on A and some specific "distinguished" elements of A. We always assume that $=$ is among the relations, but we allow the cases where there are no functions and/or no other relations and/or no distinguished elements. We use $\mathcal{A}$, $\mathcal{B}$ etc. to refer to structures. By convention the universe of a structure, unless otherwise specified, is the coresponding Latin letter.

So the simplest structures consist just of a non-empty set and $=$ on that set. Most of the structures studied in mathematics can be viewed as structures in this sense. For example, arithmetic on the integers studies the structure with universe $\mathbb{Z}$ together with $+, \cdot, -, <, =$, and perhaps 0 and 1 as distinguished elements. We normally do not mention $=$ since it is always present, and we frequently write the structure as a tuple. For example the structure just mentioned could be referred to as

$$\mathcal{A} = (\mathbb{Z}, +, \cdot, -, <, 0, 1).$$

Since $-$ is definable from $+$ ($k - n = l$ iff $l + n = k$) we could remove it from the function list without changing what can be expressed. Similarly, both 0 and 1 are inessential luxuries.

3. First Order Languages

Unlike sentential logic, there are many languages for first order logic, depending on the number and sort of functions, relations, and constants considered.

Definition 3.1. The *symbols of a first order language* $\mathcal{L}$ are as follows:

(i) some collection of symbols for functions, each of specified arity;
(ii) some collection of symbols for relations, each of specified arity and including the binary relation symbol $=$;
(iii) some collection of symbols for constants;
(iv) an infinite set of variables: $v_1, v_2, \ldots, v_n, \ldots$ for all $n \in \mathbb{N}$;
(v) the quantifiers $\forall$ and $\exists$;
(vi) the sentential connectives: $\neg, \wedge, \vee, \rightarrow$:
(vii) parentheses and comma: (,).

We emphasize that a language may contain no function symbols, no constants, and no relation symbols other than $=$. We normally use F, G, H for function symbols, P, Q, R, S for relation symbols, and c, d, e for constants. In each case, there may also be either sub- or superscripts, for example F', R^*, c_1. Note also that each function symbol and each relation symbol has a specified arity, although our notation does not show this. A language $\mathcal{L}$ is determined by the function symbols, relation symbols other than $=$, and constants that it contains. This set is frequently called the set of *non-logical symbols* of $\mathcal{L}$ and is written as $\mathcal{L}^{nl}$.

Just as we used A, B, C to refer to arbitrary atomic sentences S_i, we will x, y, and z (perhaps with sub- or superscripts) to refer to arbitrary variables v_i.

Definition 3.2. A *structure for* $\mathcal{L}$, or simply an $\mathcal{L}$*-structure*, is a structure $\mathcal{A}$ which contains an n-ary function $F^{\mathcal{A}}$ for every n-ary function symbol F of $\mathcal{L}$, an n-ary relation $R^{\mathcal{A}}$ for every n-ary relation symbol R of $\mathcal{L}$, and a distinguished element $c^{\mathcal{A}}$ for every constant c of $\mathcal{L}$, but no other functions, relations, or named elements.

For example, if the non-logical symbols of $\mathcal{L}$ are a unary function symbol F and a constant c then an $\mathcal{L}$-structure would be some $\mathcal{A} = (A, F^{\mathcal{A}}, c^{\mathcal{A}})$ where A is some non-empty set, $F^{\mathcal{A}}$ is some unary function on A, and $c^{\mathcal{A}} \in A$. Examples of such structures include $(\mathbb{Z}, s, 0)$ where $s(k) = k + 1$ for all $k \in \mathbb{Z}$, $(\mathbb{N}, f, 1)$ where $f(k) = k^2$ for all $k \in \mathbb{N}$, and $(\mathbb{Q}, g, \frac{1}{2})$ where $g(k) = \frac{k}{2}$ for all $k \in \mathbb{Q}$.

As another example, if the only non-logical symbol of $\mathcal{L}$ is a unary relation symbol P then an $\mathcal{L}$-structure would be some $\mathcal{A} = (A, P^{\mathcal{A}})$ where $P^{\mathcal{A}}$ is some unary relation on, i.e. a subset of, A. If A is finite with n elements then there are 2^n choices for $P^{\mathcal{A}}$. If A is infinite then there are infinitely many, in fact uncountably many, choices for $P^{\mathcal{A}}$.

4. Terms, Formulas, and Sentences

In sentential logic, the only meaningful expressions were the sentences. In first order logic we will also have *terms* and *formulas*. Terms will define functions (or perhaps individual elements) in any structure, and formulas will define relations in any structure. We first discuss terms via an example.

Consider the language $\mathcal{L}$ whose only non-logical symbols are a unary function symbol F and a constant c. $\mathcal{L}$-structures include $\mathcal{A} = (\mathbb{Z}, s, 0)$, $\mathcal{B} = (\mathbb{N}, f, 1)$, and $\mathcal{C} = (\mathbb{Q}, g, \frac{1}{2})$ from the preceding section. c is a term, since it defines a specific element in any $\mathcal{L}$-structure. Every variable x is a term, since it varies over the elements of the universe in any structure. F by itself is *not* a term, since there is no indication of what element the function is to be applied to. $F(c)$ is a term, and it defines the element $1 = F^{\mathcal{A}}(c^{\mathcal{A}})$ in $\mathcal{A}$, $1 = 1 \cdot 1$ in $\mathcal{B}$, and $\frac{1}{4} = \frac{1}{2} \cdot \frac{1}{2}$ in $\mathcal{C}$. $F(x)$ is a term since it defines the function $s = F^{\mathcal{A}}$ in $\mathcal{A}$, etc. Similarly $F(F(x))$ is a term; it defines the function $s(s(x)) = x + 2$ in $\mathcal{A}$, the function $f(f(x)) = x^4$ in $\mathcal{B}$, and the function $g(g(x)) = \frac{x}{4}$ in $\mathcal{C}$.

The formal definition of term is analogous to the definition of sentence in sentential logic.

Definition 4.1. The *terms of* $\mathcal{L}$ are defined as follows:

(i) every variable and every constant of $\mathcal{L}$ is a term of $\mathcal{L}$;
(ii) if $t_1, \ldots, t_n$ are terms of $\mathcal{L}$ and F is an n-ary function symbol of $\mathcal{L}$ then $F(t_1, \ldots, t_n)$ is a term of $\mathcal{L}$;
(iii) nothing else is a term of $\mathcal{L}$.

We use $Tm_{\mathcal{L}}$ for the set of all terms of $\mathcal{L}$.

For example, if F and G are binary functions symbols of $\mathcal{L}$ and c and d are constants of $\mathcal{L}$, then the following are all terms of $\mathcal{L}$:

$F(c, c)$, $G(d, c)$, $F(v_1, v_4)$, $G(v_6, F(v_6, v_4))$, $F(G(F(c, v_1), G(d, v_2)), d)$.

If $\mathcal{A}$ is the structure whose universe is $\mathbb{N}$ and in which $F^{\mathcal{A}} = +$, $G^{\mathcal{A}} = \cdot$, $c^{\mathcal{A}} = 1$, and $d^{\mathcal{A}} = 2$, then $F(c, c)$ and $G(d, c)$ both define 2, $F(v_1, v_4)$ defines $+$ (or, perhaps better, $x + y$ or even $x_1 + x_4$), and $G(v_6, F(v_6, v_4))$ defines $y \cdot (y + x)$ (or perhaps $x_6 \cdot (x_6 + x_4)$).

We next discuss formulas via an example.

Consider the language $\mathcal{L}$ whose non-logical symbols are the binary relation symbol R and a constant c. R by itself is not a formula, since it doesn't specify which elements are in the relation R. But $R(c, c)$, $R(c, v_1)$, $R(v_2, v_2)$, and $R(v_2, v_1)$ are formulas, as are $c = c$, $c = v_1$, and $v_2 = v_1$. There are called *atomic formulas* since they are not built up from other formulas. Starting with the atomic formulas we build up more complicated formulas using the connectives and quantifiers. For example $\neg R(c, c)$, $(R(v_1, v_2) \to R(v_2, v_1))$, $\exists v_2 R(v_2, v_1)$, $(\neg v_1 = c \to \exists v_2 R(v_2, v_1))$ are all formulas.

Definition 4.2. The *atomic formulas of* $\mathcal{L}$ are the expressions of the form $R(t_1, \ldots, t_n)$ where R is an n-ary relation symbol of $\mathcal{L}$ and $t_1, \ldots, t_n \in Tm_{\mathcal{L}}$. By convention we customarily write $t_1 = t_2$ instead of $= (t_1, t_2)$.

Formulas are now defined analogously to sentences in sentential logic.

Definition 4.3. The *formulas of* $\mathcal{L}$ are defined as follows:

(i) any atomic formula of $\mathcal{L}$ is a formula of $\mathcal{L}$;
(ii) if φ is a formula of $\mathcal{L}$ then so is $\neg\varphi$;
(iii) if φ and ψ are formulas of $\mathcal{L}$ then so are $(\varphi \wedge \psi)$, $(\varphi \vee \psi)$, and $(\varphi \to \psi)$;
(iv) if φ is a formula of $\mathcal{L}$ then so are $\forall v_n \varphi$ and $\exists v_n \varphi$ for all $n \in \mathbb{N}$.

We write $Fm_{\mathcal{L}}$ for the set of formulas of $\mathcal{L}$. The sequence of formulas showing how a formula φ is built up from the atomic formulas is called a *history* of φ. The formulas in a history of φ are called *subformulas* of φ.

In the formula $\exists v_2 R(v_2, v_1)$ the variables v_1 and v_2 have different roles. For example consider the structure $\mathcal{A}$ whose universe is $\mathbb{N}$ and in which R is interpreted as $<$. What this formula then says is "there is something less than v_1" — this is a property of (the elements interpreting) v_1, but *not* of v_2. The variable v_1 actually varies over the elements of the universe since it is not quantified, but v_2 does not vary since it is bound by the existential quantifier. So this formula defines a unary relation (that is, a set), in the example the set of all $k > 1$, that is the set of all numbers $k \in \mathbb{N}$ which satisfy this property when substituted for the variable v_1. The following definition makes this distinction precise.

Definition 4.4. An occurrence of a variable x in a formula φ is *bound* provided the occurrence is in a subformula of φ beginning with a quantifier on x. An occurrence which is not bound is *free*.

For example, if $\varphi = (\exists v_1 R(v_1, v_2) \vee \exists v_2 R(v_1, v_2))$ then the first two occurrences of v_1 are bound and the last occurrence is free, but the first occurence of v_2 is free and the last two are bound.

As we will see in the next section, a formula defines a relation of the (interpretations of the) variables occurring free in the formula. And a formula in which no variable occurs free will be either true or false in any interpretation.

Definition 4.5. A *sentence of* $\mathcal{L}$ is a formula in which no variable occurs free.

5. Interpretating Terms, Formulas, and Sentences in Structures

Let $t \in Tm_{\mathcal{L}}$ and let $x_1, \ldots, x_n$ list all variables occurring in t. Then t defines an n-ary function $t^{\mathcal{A}}$ in any $\mathcal{L}$-structure $\mathcal{A}$. To emphasize this fact we will sometimes write t as $t(x_1, \ldots, x_n)$. If t contains no variables then t defines an element $t^{\mathcal{A}}$ in any $\mathcal{A}$. If t is a term in the variables $x_1, \ldots, x_n$, $\mathcal{A}$ is a structure, and $a_1, \ldots, a_n \in A$ then the value of $t^{\mathcal{A}}(a_i, \ldots, a_n)$ is obtained from t by replacing each F by $F^{\mathcal{A}}$, each c by $c^{\mathcal{A}}$, and all occurrences by $x_1, \ldots, x_n$ by $a_1, \ldots, a_n$ respectively.

For example, suppose F and G are binary function symbols of $\mathcal{L}$, c is a constant of $\mathcal{L}$, and $\mathcal{A}$ is an $\mathcal{L}$-structure with universe $\mathbb{N}$ and interpreting F by $+$, G by $\cdot$, and c by 1. Let $t = G(F(v_2, c), G(v_2, F(c, c)))$. Then $t^{\mathcal{A}}$ is a unary function and for any $k \in \mathbb{N}$ we have $t^{\mathcal{A}}(k) = (k+1) \cdot (k \cdot (1+1))$ or simply $(k+1) \cdot k \cdot 2$.

With the same $\mathcal{L}$ and $\mathcal{A}$, consider the term $t = F(G(v_3, v_3), G(v_3, v_4))$. Then $t^{\mathcal{A}}$ is a binary function and $t^{\mathcal{A}}(k, l) = k \cdot k + k \cdot l$ for every $k, l \in \mathbb{N}$.

Let $\varphi \in Fm_{\mathcal{L}}$ and let $x_1, \ldots, x_n$ list all of the variables which occur free in φ. Then φ defines an n-ary relation $\varphi^{\mathcal{A}}$ in any $\mathcal{L}$-structure $\mathcal{A}$. We will frequently write φ as $\varphi(x_1, \ldots, x_n)$ in this case. The following definition of *satisfaction* shows both how $\varphi^{\mathcal{A}}$ is defined and how the truth of a sentence in $\mathcal{A}$ is defined.

Definition 5.1. Let $\mathcal{A}$ be an $\mathcal{L}$-structure. We define the relation $a_1, \ldots, a_n$ *satisfies* φ *in* $\mathcal{A}$ for $\varphi(x_1, \ldots, x_n) \in Fm_{\mathcal{L}}$ and $a_1, \ldots, a_n \in A$ as follows:

(i) If φ is $R(t_1, \ldots, t_k)$ then $a_1, \ldots, a_n$ satisfies φ in $\mathcal{A}$ iff
$R^{\mathcal{A}}(t_1^{\mathcal{A}}(a_1, \ldots, a_n), \ldots, t_k^{\mathcal{A}}(a_1, \ldots, a_n))$ holds;
if φ is $t_1 = t_2$ then $a_1, \ldots, a_n$ satisfies φ in $\mathcal{A}$ provided $t_1^{\mathcal{A}}(a_1, \ldots, a_n) = t_2^{\mathcal{A}}(a_1, \ldots, a_n)$.

(ii) If φ is $\neg\psi$ then $a_1, \ldots, a_n$ satisfies φ in $\mathcal{A}$ iff $a_1, \ldots, a_n$ does *not* satisfy ψ in $\mathcal{A}$.

(iii) The cases $\varphi = (\psi * \theta)$ for a binary connective $*$ are also defined just as in sentential logic.

(iv) If φ is $\forall y \psi$ for some formula $\psi(x_1, \ldots, x_n, y)$ then $a_1, \ldots, a_n$ satisfies φ in $\mathcal{A}$ iff $a_1, \ldots, a_n, b$ satisfies ψ in $\mathcal{A}$ for *every* $b \in A$; the case $\varphi = \exists y \psi$ is handled analogously.

Notation: We write $\mathcal{A} \models \varphi(a_1, \ldots, a_n)$ if $a_1, \ldots, a_n$ satisfies φ in $\mathcal{A}$. If φ is a sentence we write $\mathcal{A} \models \varphi$, also read $\mathcal{A}$ *models* φ, if the empty sequence satisfies φ in $\mathcal{A}$.

The relation $\varphi^{\mathcal{A}}$ can now be defined as follows:
$\varphi^{\mathcal{A}}(a_1, \ldots, a_n)$ holds iff $\mathcal{A} \models \varphi(a_1, \ldots, a_n)$.

WARNING: $\varphi(a_1, \ldots, a_n)$ is *not* a formula of $\mathcal{L}$ for $n \geq 1$ since $a_1, \ldots, a_n$ are not symbols of $\mathcal{L}$.

We look at some examples where $\mathcal{L}^{nl} = \{R\}$ for a binary relation symbol R. Consider first the structure $\mathcal{A} = (\mathbb{N}, <)$. No $k \in \mathbb{N}$ satisfies $R(v_1, v_1)$ since $k < k$ always fails. Therefore every k satisfies $\neg R(v_1, v_1)$ and hence $\mathcal{A} \models \forall v_1 \neg R(v_1, v_1)$.

k satisfies $\exists y R(y, x)$ in $\mathcal{A}$ iff there is some $l \in \mathbb{N}$ such that $\mathcal{A} \models R(l, k)$, i.e. $l < k$, which holds iff $k > 1$. Therefore the formula $\neg\exists y R(y, x)$ defines the set $\{1\}$.

k_1, k_2 satisfies $\exists y (R(v_1, y) \wedge R(y, v_2))$ iff there is some $l \in \mathbb{N}$ with $\mathcal{A} \models R(k_1, l) \wedge R(l, k_2)$, that is, iff $k_1 + 1 < k_2$. Therefore $\neg\exists y (R(v_1, y) \wedge R(y, v_2))$ defines the relation which holds between k_1, k_2 iff $k_1 + 1 = k_2$, i.e. k_2 is the immediate successor of k_1.

The reader should consider these examples also for the structures $\mathcal{B} = (\mathbb{N}, \leq)$, $\mathcal{C} = (\mathbb{Z}, <)$, $\mathcal{D} = (\mathbb{Q}, <)$, and $\mathcal{E} = (\mathbb{N}, >)$. In particular, show that $\{1\}$ is also definable in $\mathcal{B}$ and $\mathcal{E}$, but not by the same formula as in $\mathcal{A}$.

6. Validity, Satisfiability, and Logical Consequence

We define validity and satisfiability of sentences just as we did for sentential logic.

Definition 6.1. A sentence θ of $\mathcal{L}$ is *logically true*, or *valid*, iff $\mathcal{A} \models \theta$ for every $\mathcal{L}$-structure $\mathcal{A}$. We write $\models \theta$ to mean that θ is valid.

Definition 6.2. A sentence θ of $\mathcal{L}$ is *satisfiable* iff $\mathcal{A} \models \theta$ for some $\mathcal{L}$-structure $\mathcal{A}$.

The following important fact is proved exactly like the corresponding result for sentential logic (see Lemma 4.1 in Chapter 1).

Lemma 6.1. *The sentence θ is satisfiable iff $\neg\theta$ is not valid; θ is valid iff $\neg\theta$ is not satisfiable.*

Unfortunately there is no method like truth tables to check the satisfiability or validity of a sentence. Sometimes in checking whether or not a sentence is valid it helps to suppose that there is some $\mathcal{A}$ falsifying it and work backwards to see what $\mathcal{A}$ must look like.

Example. Decide whether or not θ defined as $\forall x(P(x) \to Q(x)) \to (\forall x P(x) \to \forall x Q(x))$ is valid. The definition of satisfaction yields the following equivalences:

$\mathcal{A} \not\models \theta$ iff

$\mathcal{A} \models \forall x(P(x) \to Q(x))$ and $\mathcal{A} \not\models \forall x P(x) \to \forall x Q(x)$ iff

$\mathcal{A} \models \forall x(P(x) \to Q(x))$, $\mathcal{A} \models \forall x P(x)$, and $\mathcal{A} \not\models \forall x Q(x)$ iff

$\mathcal{A} \models P(a) \to Q(a)$ for all $a \in A$, $\mathcal{A} \models P(a)$ for all $a \in A$, and $\mathcal{A} \models \neg Q(b_0)$ for some $b_0 \in A$ iff

(either $\mathcal{A} \models \neg P(a)$ or $\mathcal{A} \models Q(a)$) for all $a \in A$, $\mathcal{A} \models P(a)$ for all $a \in A$, and $\mathcal{A} \models \neg Q(b_0)$ for some $b_0 \in A$.

But it is impossible to satisfy this last condition, so there can be no such $\mathcal{A}$ and so θ is valid.

Example Decide whether or not θ defined as $\exists x(P(x) \to Q(x)) \to (\exists x P(x) \to \exists x Q(x))$ is valid. We obtain the following sequence of equivalences:

$\mathcal{A} \not\models \theta$ iff

$\mathcal{A} \models \exists x(P(x) \to Q(x))$, $\mathcal{A} \models \exists x P(x)$, and $\mathcal{A} \not\models \exists x Q(x)$ iff

$\mathcal{A} \models P(a_0) \to Q(a_0)$ for some $a_0 \in A$, $\mathcal{A} \models P(a_1)$ for some $a_1 \in A$, and $\mathcal{A} \models \neg Q(b)$ for all $b \in A$.

But the conditions in this last line can be satisfied, for example by defining $A = \{0, 1\}$, $P^{\mathcal{A}} = \{1\}$, and $Q^{\mathcal{A}} = \emptyset$. Thus θ is not valid, since it is false on the structure $\mathcal{A}$ we just defined.

It is convenient to define validity also for formulas with free variables. Let $\varphi \in Fm_{\mathcal{L}}$ have free variables $x_1, \ldots, x_n$, and let $\mathcal{A}$ be an $\mathcal{L}$-structure. We say that φ is *true* on $\mathcal{A}$, notation $\mathcal{A} \models \varphi$, provided that $\mathcal{A} \models \forall x_1, \ldots, \forall x_n \varphi$. Note that, unlike sentences, usually neither φ nor $\neg\varphi$ is true on $\mathcal{A}$.

Definition 6.3. A formula φ of $\mathcal{L}$ is *valid* iff $\mathcal{A} \models \varphi$ for every $\mathcal{L}$-structure $\mathcal{A}$. We write $\models \varphi$ to mean that φ is valid.

In the first example above we showed that
$\models \forall x(P(x) \to Q(x)) \to (\forall x P(x) \to \forall x Q(x))$.
A similar argument shows that for any formulas φ and ψ we have
$\models \forall x(\varphi \to \psi) \to (\forall x\varphi \to \forall x\psi)$.

Satisfiability of sets of sentences and logical consequence are also defined just as in sentential logic.

Definition 6.4. A set Σ of sentences of $\mathcal{L}$ is *satisfiable* iff there is some $\mathcal{A}$ such that $\mathcal{A} \models \Sigma$ (meaning $\mathcal{A} \models \sigma$ for every $\sigma \in \Sigma$).

Definition 6.5. Let Σ be a set of sentences of $\mathcal{L}$ and let $\varphi \in Fm_{\mathcal{L}}$. Then φ is a *logical consequence* of Σ, written $\Sigma \models \varphi$, iff $\mathcal{A} \models \varphi$ for every $\mathcal{A} \models \Sigma$.

We also obtain the following just as in sentential logic.

Lemma 6.2. *Assume that $\Sigma \subseteq Sn_{\mathcal{L}}$ and $\theta \in Sn_{\mathcal{L}}$. Then $\Sigma \models \theta$ iff $(\Sigma \cup \{\neg\theta\})$ is not satisfiable.*

We also have the expected notion of *equivalence* of formulas.

Definition 6.6. Formulas φ and ψ are *equivalent*, written $\varphi \equiv \psi$, iff $\models \varphi \to \psi$ and $\models \psi \to \varphi$.

For example, just as in sentential logic, we see the following:
$(\varphi \vee \psi) \equiv (\neg\varphi \to \psi)$ and
$(\varphi \wedge \psi) \equiv \neg(\varphi \to \neg\psi)$.
Furthermore, using the definition of satisfaction, we also obtain:
$\exists x\varphi \equiv \neg\forall x \neg\varphi$.

We thus obtain the following result:

Theorem 6.1. *For any $\varphi \in Fm_{\mathcal{L}}$ there is some $\varphi^* \in Fm_{\mathcal{L}}$ such that $\varphi \equiv \varphi^*$ and φ^* is built using just $\neg$, $\to$ and $\forall$.*

7. Properties of Validity and Logical Consequence

In this section we identify some classes of valid formulas and some properties of logical consequence which will, in particular, be used in defining our system of deduction.

Tautologies. Let θ be a sentence of sentential logic which is a tautology, for example, $A \to (B \to A)$. Let θ^* result by replacing each atomic sentence in θ by a formula of $\mathcal{L}$, for example replacing A by $P(x)$ and replacing B by $Q(x)$, so θ^* is $P(x) \to (Q(x) \to P(x))$. Then θ^* is a formula of $\mathcal{L}$ and $\models \theta^*$. We will also call the formula θ^* a *tautology*.

Universal Quantification. (a) Assume that $\Sigma \models \varphi$, where $\Sigma \subseteq Sn_{\mathcal{L}}$ and $\varphi \in Fm_{\mathcal{L}}$. Then for any variable x we have $\Sigma \models \forall x\varphi$. If x does not occur free in φ this is clear, since in this case $\varphi \equiv \forall x\varphi$. If x does occur free in φ we obtain this since $\mathcal{A} \models \varphi$ means that $\mathcal{A} \models \forall x_1 \dots \forall x_n \varphi$ and x is some x_i.

(b) If x does not occur free in φ then $\models (\varphi \to \forall x\varphi)$ since, as remarked above, $\varphi \equiv \forall x\varphi$ under this condition. The assumption that x does not occur free in φ is essential since, for example, $(P(x) \to \forall x P(x))$ is not valid.

Modus Ponens. Assume that $\Sigma \models \varphi$ and $\Sigma \models (\varphi \to \psi)$. Then $\Sigma \models \psi$. Let $x_1, \dots, x_n$ list all variables occurring free in either φ or ψ, so we can write $\varphi(x_1, \dots, x_n)$ and $\psi(x_1, \dots, x_n)$. Let $\mathcal{A} \models \Sigma$. Then $\mathcal{A} \models \varphi(a_1, \dots, a_n)$ and $\mathcal{A} \models \varphi(a_1, \dots, a_n) \to \psi(a_1, \dots, a_n)$ for all $a_1, \dots, a_n \in A$. So by the definition of satisfaction we must have $\mathcal{A} \models \psi(a_1, \dots, a_n)$ for all $a_1, \dots, a_n \in A$, and so $\mathcal{A} \models \psi$.

Substitution. Suppose that φ has only x free, so we can write φ as $\varphi(x)$. If t is any term, then the formula obtained by replacing all free occurrences of x in φ by t will be written as $\varphi(t)$. We "expect" that $(\forall x\varphi(x) \to \varphi(t))$ should be valid since, intuitively, $\forall x\varphi$ says that every element of the universe satisfies φ and $\varphi(t)$ says that the element named by t satisfies φ. If t is a constant c then this is exactly correct, since $\mathcal{A} \models \forall x\varphi(x)$ implies that $\mathcal{A} \models \varphi(a)$ for all $a \in A$, in particular for $a = c^{\mathcal{A}}$.

But if t is a variable y (or contains a variable) this can fail — let $\varphi(x)$ be $\exists y R(x, y)$; then $\varphi(y)$ is $\exists y R(y, y)$ and it is easy to see that $(\forall x \exists y R(x, y) \to \exists y R(y, y))$ is not valid. But this is the only difficulty — if no new occurrence of y in $\varphi(y)$ is bound (for example if y does not occur bound in $\varphi(x)$) then $\models (\forall x\varphi(x) \to \varphi(y))$. More generally, $\models (\forall x\varphi(x) \to \varphi(t))$ provided no new occurrence in $\varphi(t)$ of a variable in t is bound.

Finally, if φ also has $z_1, \dots, z_k$ free, so φ can be written as $\varphi(x, z_1, \dots, z_k)$, then we have $\models (\forall x\varphi(x, z_1, \dots, z_k) \to \varphi(t, z_1, \dots, z_k))$ under the same circumstances.

8. A Formal Proof System

In this section we define a deductive system such that the deducible formulas are precisely the valid formulas — this is the Completeness Theorem, proved in the next Chapter. As with sentential logic we first specify a set of formulas called the *logical axioms*, and the only rule we use is modus ponens. A formula is *deducible* iff it is obtained from the axioms by a finite number of applications of modus ponens. We restrict to formulas using only $\neg$, $\to$, and $\forall$. There is no loss in doing so since we know by Theorem 6.1 that every formula is logically equivalent to such a formula.

Definition 8.1. The set $\Lambda_{\mathcal{L}}$ of *(logical) axioms* of $\mathcal{L}$ consists of all formulas of the following forms:

(1) $\forall y_1 \dots \forall y_n \varphi$ where φ is a tautology,
(2) $\forall y_1 \dots \forall y_n \big(\forall x\varphi(x, \dots) \to \varphi(t, \dots)\big)$ where no new occurrence in $\varphi(t, \dots)$ of a variable in t is bound,
(3) $\forall y_1 \dots \forall y_n \big(\forall x(\varphi \to \psi) \to (\forall x\varphi \to \forall x\psi)\big)$,

(4) $\forall y_1 \ldots \forall y_n(\varphi \to \forall x\varphi)$ where x does not occur free in φ,
(5) axioms for equality which we will give later.

Lemma 8.1. *If $\varphi \in \Lambda_{\mathcal{L}}$ then $\models \varphi$.*

Definition 8.2. A *(logical) deduction* is a finite sequence $\varphi_1, \ldots, \varphi_n$ of formulas of $\mathcal{L}$ such that for every $i \leq n$ one of the following holds:

(i) $\varphi_i \in \Lambda_{\mathcal{L}}$,
(ii) there are $j, k < i$ such that φ_i follows from φ_j and φ_k by modus ponens, that is, $\varphi_k = (\varphi_j \to \varphi_i)$.

Definition 8.3. A formula φ is *(logically) deducible*, notation $\vdash \varphi$, iff there is a deduction whose last formula is φ.

As in sentential logic, every axiom is deducible by a deduction of length one, and any non-trivial deduction has length at least three. Here is an example, where $\mathcal{L}$ contains a unary relation symbol P:

1. $\forall x \neg P(x) \to \neg P(x)$ – axiom (2)
2. $(\forall x \neg P(x) \to \neg P(x)) \to (P(x) \to \neg \forall x \neg P(x))$ – axiom (1) where the tautology is $(A \to \neg B) \to (B \to \neg A)$
3. $(P(x) \to \neg \forall x \neg P(x))$

This establishes that $\vdash (P(x) \to \neg \forall x \neg P(x))$, which may look more familiar as $\vdash (P(x) \to \exists x P(x))$.

We can show that every deducible formula is valid exactly as in sentential logic (see Theorem 8.1 in Chapter 1).

Theorem 8.1. *If $\vdash \varphi$ then $\models \varphi$.*

We also use deductions from hypotheses.

Definition 8.4. Let $\Gamma \subseteq Fm_{\mathcal{L}}$. Then a *deduction from* Γ is a finite sequence $\varphi_1, \ldots, \varphi_n$ of formulas of $\mathcal{A}$ such that for every $i \leq n$ one of the following holds:

(i) $\varphi_i \in (\Gamma \cup \Lambda_{\mathcal{L}})$,
(ii) there are $j, k < i$ such that φ_i follows from φ_j and φ_k by modus ponens.

We say that φ *is deducible from* Γ, notation $\Gamma \vdash \varphi$, iff there is a deduction from Γ whose last formula is φ.

We have the following fact, corresponding to the rule of modus ponens.

Lemma 8.2. *Let $\Gamma \subseteq Fm_{\mathcal{L}}$ and let $\varphi, \psi \in Fm_{\mathcal{L}}$. Assume that $\Gamma \vdash \varphi$ and $\Gamma \vdash (\varphi \to \psi)$. Then $\Gamma \vdash \psi$.*

We have soundness for deductions from sets of sentences.

Theorem 8.2. *Let $\Sigma \subseteq Sn_{\mathcal{L}}$ and let $\varphi \in Fm_{\mathcal{L}}$. If $\Sigma \vdash \varphi$ then $\Sigma \models \varphi$.*

9. The Deduction Theorem and Other Derived Rules

We prove several "derived rules" which will help in showing formulas are deducible. The first of these is the Deduction Theorem, which is stated and proved just as it was for sentential logic.

Theorem 9.1. *(The Deduction Theorem) Assume that* $(\Gamma \cup \{\varphi\}) \vdash \psi$. *Then* $\Gamma \vdash (\varphi \rightarrow \psi)$.

Just as in sentential logic, we use the Deduction Theorem to reduce the problem of showing that $\Gamma \vdash (\varphi \rightarrow \psi)$ to the simpler problem of showing that $(\Gamma \cup \{\varphi\}) \vdash \psi$.

The various forms of Generalization are similarly used to simplify the problem of showing that $\Gamma \vdash \forall x\varphi$.

Theorem 9.2. *(Generalization) Assume that the variable x does not occur free in any formula in Γ. Assume that $\Gamma \vdash \varphi$. Then $\Gamma \vdash \forall x\varphi$.*

Proof. (Outline) Let $\varphi_1 \ldots, \varphi_n$ be a deduction from Γ of φ. We show that $\Gamma \vdash \forall x\varphi_i$ for all $i \leq n$ by induction on i. If $\varphi_i \in \Lambda_{\mathcal{L}}$ then also $\forall x\varphi_i \in \Lambda_{\mathcal{L}}$. If $\varphi_i \in \Gamma$ then by hypothesis x cannot occur free in φ_i, so $(\varphi_i \rightarrow \forall x\varphi) \in \Lambda_{\mathcal{L}}$ and thus $\Gamma \vdash \forall x\varphi_i$ by modus ponens. The inductive step uses a logical axiom of form (3). □

Note that the hypothesis is automatically satisfied if $\Gamma = \emptyset$ or Γ is a set of sentences.

We note without proof the following variation on Generalization.

Theorem 9.3. *(Generalization on Constants) Let $\Gamma \subseteq Fm_{\mathcal{L}}$ and let $\varphi(x) \in Fm_{\mathcal{L}}$. Assume that the constant c does not occur in any formula in Γ and does not occur in $\varphi(x)$. Assume that $\Gamma \vdash \varphi(c)$. Then $\Gamma \vdash \forall x\varphi(x)$.*

There are several special results useful for showing that negations are deducible. They are easily proved by referring to the appropriate tautology. Note that the converses also hold, but this is usually not useful in deductions.

Lemma 9.1. (a) *If $\Gamma \vdash \varphi$ then $\Gamma \vdash \neg\neg\varphi$.*
(b) *If $\Gamma \vdash \varphi$ and $\Gamma \vdash \neg\psi$ then $\Gamma \vdash \neg(\varphi \rightarrow \psi)$.*
(c) *(Contraposition) If $(\Gamma \cup \{\varphi\}) \vdash \psi$ then $(\Gamma \cup \{\neg\psi\}) \vdash \neg\varphi$.*

Consistent sets are defined as in sentential logic and have the same properties with the same proofs, which we therefore omit.

Definition 9.1. A set Γ of formulas is *consistent* iff there is no formula φ such that $\Gamma \vdash \varphi$ and $\Gamma \vdash \neg\varphi$. Γ is *inconsistent* if it is not consistent.

Lemma 9.2. *If Γ is inconsistent then $\Gamma \vdash \psi$ for all formulas ψ.*

Theorem 9.4. *(Proof by Contradiction) $\Gamma \vdash \varphi$ if (and only if) $(\Gamma \cup \{\neg\varphi\})$ is inconsistent. $\Gamma \vdash \neg\varphi$ if (and only if) $(\Gamma \cup \{\varphi\})$ is inconsistent.*

We illustrate the use of these rules in the following example.

Example. Show that $\vdash \exists x \forall y R(x,y) \to \forall y \exists x R(x,y)$.
Eliminating the existential quantifier we see that what we need to do is to show that $\vdash \neg\forall x \neg\forall y R(x,y) \to \forall y \neg\forall x \neg R(x,y)$.
By the Deduction Theorem it suffices to show $\neg\forall x\neg\forall y R(x,y) \vdash \forall y\neg\forall x\neg R(x,y)$.
By Generalization, since y is not free on the left of $\vdash$, it suffices to show $\neg\forall x\neg\forall y R(x,y) \vdash \neg\forall x\neg R(x,y)$.
By Contraposition it suffices to show $\forall x\neg R(x,y) \vdash \forall x\neg\forall y R(x,y)$.
By Generalization, since x is not free on the left of $\vdash$, it suffices to show $\forall x\neg R(x,y) \vdash \neg\forall y R(x,y)$.
By Proof by Contradiction it suffices to show that $\Gamma = \{\forall x\neg R(x,y), \forall y R(x,y)\}$ is inconsistent, which is shown by the following deduction from Γ.

1. $\forall x\neg R(x,y)$ — hypothesis
2. $\forall y R(x,y)$ — hypothesis
3. $\forall x\neg R(x,y) \to \neg R(x,y)$ — axiom (2)
4. $\forall y R(x,y) \to R(x,y)$ — axiom (2)
5. $\neg R(x,y)$ — MP on lines 1, 3
6. $R(x,y)$ — MP on lines 2, 4

10. Expressability

What can we say about a structure using sentences of first order logic? What can we define in a structure using formulas of first order logic? We give some examples for the language $\mathcal{L}$ whose only non-logical symbol is a binary relation symbol R.

An important class of $\mathcal{L}$-structures is the class of *linear orders*. This includes the familiar structures $(\mathbb{N},<)$, $(\mathbb{Z},<)$, $(\mathbb{Q}\ <,)$, and $(\mathbb{R},<)$. The class of linear orders can be axiomatized by the set $\Sigma_{l.o.}$ containing the following $\mathcal{L}$-sentences:

$\forall x\neg R(x,x)$,
$\forall x\forall y\forall z\big((R(x,y)\wedge R(y,z)) \to R(x,z)\big)$,
$\forall x\forall y\big(R(x,y)\vee x=y\vee R(y,x)\big)$.

Other properties of linear orders follow from these three axioms. For example, $\Sigma_{l.o.} \models \forall x\forall y\big(R(x,y)\to\neg R(y,x)\big)$.

This definition of linear order gives what are sometimes called *strict* linear orders – non-strict linear orders include $(\mathbb{N},\leq)$, etc.

Linear orders may differ on other sentences of $\mathcal{L}$ — for example, any two of $(\mathbb{N},<)$, $(\mathbb{Z},<)$, and $(\mathbb{Q},<)$ can be distinguished by a sentence of $\mathcal{L}$. However, $(\mathbb{Q},<)$ and $(\mathbb{R},<)$ both satisfy precisely the same sentences of $\mathcal{L}$. In fact, if we define $\Sigma_{d.l.o.}$ to be $\Sigma_{l.o.}$ together with the sentences $\forall x\exists y R(x,y)$, $\forall y\exists x R(x,y)$, and $\forall x\forall y\big(R(x,y)\to\exists z(R(x,z)\wedge R(x,y))\big)$ then we have the following remarkable result, whose proof is beyond the scope of these notes.

Theorem 10.1. *For every $\theta \in Sn_{\mathcal{L}}$ either $\Sigma_{d.l.o.} \models \theta$ or $\Sigma_{d.l.o.} \models \neg\theta$.*

Corollary 10.1. *Let $\mathcal{A}$ and $\mathcal{B}$ be $\mathcal{L}$-structures and assume that they both model $\Sigma_{d.l.o.}$. Then for every $\theta \in Sn_{\mathcal{L}}$ we have $\mathcal{A}\models\theta$ iff $\mathcal{B}\models\theta$.*

If we add to $\Sigma_{l.o.}$ sentences saying "there is a first element, every element has an immediate successor, and every element except the first has an immediate predecessor" we obtain the set $\Sigma_{n.l.o.}$ which axiomatizes the sentences true on $(\mathbb{N}, <)$.

Theorem 10.2. *For every $\theta \in Sn_{\mathcal{L}}$ we have $(\mathbb{N}, <) \models \theta$ iff $\Sigma_{n.l.o.} \models \theta$.*

Of course there are other structures which are also models of $\Sigma_{n.l.o.}$, for example $(\mathbb{N} \cup \{0\}, <)$ and $(\{k \in \mathbb{Z} : k < 0\}, >)$. More importantly, there are also models $\mathcal{A}$ of $\Sigma_{n.l.o.}$ which contain "infinite" elements, that is, elements a such that $\{b \in A : R^{\mathcal{A}}(b, a) \text{ holds}\}$ is infinite. This will be a consequence of the Completeness Theorem proved in the next Chapter.