

# Lectures On Mathematical Logic

## Lecture One - Languages and Structures

Amir H. Ebrahimnezhad

University of Tehran Department of Physics.

July 22, 2023

### Preface

The Lecture notes before you are followed from courses and books that have been read by me. I would try to work and update this every once in a while. This edition is updated until July 22, 2023. For any question or corrections please contact me *Thisis-meamir@outlook.com*.

### Formal Languages

In logic, a formal language consists of words whose letters are taken from an alphabet and are well-formed according to a specific set of rules. In this context we construct a restricted formal language, and our goal is to be able to form a certain statements about certain kinds of mathematical structures.

**Definition 1** –A **First-Order Language**  $\mathcal{L}$  is an infinite collection of distinct symbols, no one of which is properly contained in another, separated into the following categories:

1. Parantheses:  $(, )$ .
2. Connectives:  $\vee, \neg$
3. Quantifier:  $\forall$
4. Variables: One for each natural number  $\rightarrow v_n$
5. Equality symbol:  $=$
6. Constant symbols: Some set of zero or more symbols.
7. Function symbols: For each positive integer  $n$ , some set of zero or more  $n$ -ary functions symbols.
8. Relation symbols: For each positive integer  $n$ , some set of zero or more  $n$ -ary relation symbols.

**Callout** — To say a function or a relation is  $n$ -ary, means that it is intended to represent a function or relation of  $n$  variables.

**Callout** — To specify a language, all we have to do is to determine which, if any, constant, function, and relation symbols we wish to use.

**§ Formulas and Terms:** Suppose a language  $\mathcal{L} = \{v, +, <\}$ . Any combination of these elements are said to be a word in that lan-

guage, but not all the combinations are pleasing. For instance:

$$v + v < (v + v) + v$$

and

$$v + + < v$$

are both words in the language we made but one would argue that the latter has no meaning and thus is of no interest for us. Therefore we would start defining some meaningful words for our language:

**Definition 2** –If  $\mathcal{L}$  is a language, a **term of**  $\mathcal{L}$  is a nonempty finite string  $t$  of symbols from  $\mathcal{L}$  such that either:

1.  $t$  is a variable, or
2.  $t$  is a constant symbol, or
3.  $t \equiv f t_1 t_2 t_3 \dots t_n$  where  $f$  is an  $n$ -ary function symbol of  $\mathcal{L}$  and each of the  $t_i$  is a term of  $\mathcal{L}$

**Callout** — The symbol  $\equiv$  is not a part of strings of the language  $\mathcal{L}$ . Rather it is a meta-linguistic symbol that means that the strings of  $\mathcal{L}$ -symbols on each side of it are identical.

The terms of  $\mathcal{L}$  play the role of the nuons of the language. To make meaningful mathematical statements about some mathematical structure, we will want to be able to make assertions about the objects of the structure. These assertions will be the formulas of  $\mathcal{L}$ :

**Definition 3** –If  $\mathcal{L}$  is a first-order language, a **formula of**  $\mathcal{L}$  is a noneempty finite string  $\phi$  of symbols from  $\mathcal{L}$  such that either:

1.  $\phi \equiv t_1 t_2$  where  $t_i$  is a term in  $\mathcal{L}$ , or
2.  $\phi \equiv R t_1 t_2 \dots t_n$  where  $R$  is an  $n$ -ary relation and  $t_i$ s are terms in  $\mathcal{L}$ , or
3.  $\phi \equiv (\neg \alpha)$  where  $\alpha$  is a formula of  $\mathcal{L}$ , or
4.  $\phi \equiv (\alpha \vee \beta)$ , where  $\alpha, \beta$  are formulas of  $\mathcal{L}$
5.  $\phi \equiv (\forall v)(\alpha)$ , where  $v$  is a variable and  $\alpha$  is a formula of  $\mathcal{L}$ .

If a formula  $\psi$  contains the subformula  $\forall v \alpha$  [meaning that the string of symbols that constitute the formula is a substring of the string of symbols that make up  $\psi$ ], we will say that the **scope** of the quantifier  $\forall$  is  $\alpha$ . Any symbols in  $\alpha$  will be said to lie within the scope of the quantifier  $\forall$ .

**Callout** — Notice that a formula can have several different occurrences of the symbol  $\forall$ , and each occurrence of the quantifier will have its own scope.

The **Atomic Formulas** of  $\mathcal{L}$  are those formulas that satisfy clause (1) or (2) of the definition. Also note that the terms of a language are not formulas. In addition to the similarities between terms and nouns, formulas are statements in the language. Statements unlike nouns can be true or false.

§ **Induction:** We already used induction in many high school mathematics. It is now our goal to discuss the proofs by induction using the language, formulas, and terms we learned and generalize that notion of induction to a setting that will allow us to use induction to prove things about terms and formulas rather than just the natural numbers:

In the inductive step of the proof, we show the following implication:

If the formula holds for  $k$ , then the formula holds for  $k + 1$

We prove this implication by assuming the antecedent, that the theorem holds for a number  $k$ , and from that assumption we show that it also holds for  $k + 1$ . Notice that the formula itself is universal, means it is for all  $k$ s but here we only assume a certain  $k$ , therefore we are not assuming the theorem itself. Looking at this from another way we've shown that for a set  $S$  of numbers that holds the theorem true, there is these facts:

1. The number 1 is an element of  $S$ . We prove this explicitly in the base case of the proof.
2. If the number  $k$  is an element of  $S$ , then the number  $k + 1$  is also an element of  $S$ . (This is the inductive step)

But now, notice that we know that the collection of natural numbers can be defined as the smallest set such that the facts above imply, but it is not always the largest. So  $S$ , the collection of numbers which the theorem holds, is identical with the set of natural numbers, thus the theorem holds for every natural number  $n$ , as needed. So what makes the proof by induction work is that the natural numbers can be defined recursively. This is an important thing!

Looking at the definition of the formula the first two clauses has the role of (1)st fact about the set  $S$  we had here. They are explicitly defined. The last three clauses are the recursive cases, showing how if  $\alpha, \beta$  are a formula, we can generate more formulas with those. Now since the collection of formulas is defined recursively we can use an inductive-style proof when we want to prove that something is true about every formula! The proof is consist of showing that (1) The theorem holds for every atomic formula. (2) In the inductive phase of the proof, we assume that the theorem is true about simple formulas  $(\alpha, \beta)$  and use that assumption to prove that the theorem holds a more complicated formula  $\phi$  that is generated by a recursive clause of the definition. This method of proof is called *Induction on the complexity of the formula*, or *Induction on the structure of the formula*.

§ **Sentences:** Assume the language of number theory  $\mathcal{L}_{NT} = \{0, S, +, \cdot, E, <\}$ , and consider the following formulas:

$$\neg(\forall x)[(x < y) \vee (x = y)]$$

$$(\forall x)(\forall y)[(x < y) \vee (x = y) \vee (y < x)]$$

The second formula is true, for all of the replacements of  $x$  and for all  $y$ , but the truth of the first statement, is rather indetermined. For some numbers of  $x, y$  the formula holds and for others it does not. The formulas of  $\mathcal{L}$  fall into two classes: The Sentences such as the second formula, and the non-sentences. To begin we would start from defining a free-variable.

We define a free-variable as a variable that the truth of a formula depends on it. More precisely:

**Definition 4** –Suppose that  $v$  is a variable and  $\phi$  is a formula. We will say that  $v$  is **free** in  $\phi$  if:

1.  $\phi$  is atomic and  $v$  occurs in  $\phi$ , or
2.  $\phi \equiv (\neg\alpha)$  and  $v$  is free in  $\alpha$ , or
3.  $\phi \equiv (\alpha \vee \beta)$  and  $v$  is free in either  $\alpha$  or  $\beta$ , or
4.  $\phi \equiv (\forall u)(\alpha)$  and  $v$  is not  $u$  and is free in  $\alpha$ .

And so:

**Definition 5** –A **Sentence** in a language  $\mathcal{L}$  is a formula of  $\mathcal{L}$  that contains no free variable.

## Structures

For the language of number thoery, as an exmaple, it is certainly true that you know an interpretation for the symbols in that language, where  $S$  is the map that takes  $x$  to  $x + 1$  and  $E$  is exponentiation function, 0 represents number zero and so on. The point here is that there are many different possible interpretations for these symbols, and we want to be able to specify which of those interpretations is the one we are talking about. There is nothing sacred about oe mathematical structure as opposed to another. Without determining the structure under consideration, without deciding how we wish to interpret the symbols of the language, we have no way of talking about the truth or falsity of any sentence, even the trivial ones:

**Definition 6** –Fix a language  $\mathcal{L}$ . An  $\mathcal{L}$ -Structure  $\mathcal{A}$  is a nonempty set  $A$ , called **the universe of  $\mathcal{A}$** , together with:

1. For each constant symbol  $c$  of  $\mathcal{L}$ , an element  $c^{\mathcal{A}}$  of  $A$ .
2. For each  $n$ -ary function symbol of  $\mathcal{L}$ , a function  $f^{\mathcal{A}} : A^n \rightarrow A$
3. For each  $n$ -ary relation symbol of  $\mathcal{L}$ , an  $n$ -ary relation  $R^{\mathcal{A}}$  on  $A$ .

§ **Truth in a Structure** We have some formal rules about what constitutes a language, and we can identify the terms, formulas, and sentences of a language. We can also identify  $\mathcal{L}$ -structures for a give language  $\mathcal{L}$ . Now we will decide what is means to say that an  $\mathcal{L}$ -formula is true in an  $\mathcal{L}$ -structure.

**Definition 7** –If  $\mathcal{A}$  is an  $\mathcal{L}$ -structure, a **variable assignment function into  $\mathcal{A}$**  is a function  $s$  that assigns to each variable and element of the universe  $A$ . So a variable assignment function into  $\mathcal{A}$  is any function with domain  $Vars$  and Codomain  $A$ .

These assignment functions will formalize what it means to interpret a term or a formula in a structure. We will occasionally want to fix the value of the assignment function  $s$  for a certain set of variables:

**Definition 8** –If  $s$  is a variable assignment function into  $\mathcal{A}$  and  $x$  is a variable and  $a \in A$ , then  $s[x|a]$  is the variable assignment function into  $\mathcal{A}$  defined as:

$$s[x|a] = \begin{cases} s(v) & \text{if } v \text{ is a variable other than } x. \\ a & \text{if } v \text{ is the variable } x. \end{cases} \quad (1)$$

We call this function an  **$x$ -modification of the assignment function  $s$** . It is the same as  $s$ , but a certain variable  $x$  is assigned to a certain element of the universe  $A$ .

Now we would generalize this to a term assignment function.

**Definition 9** –Suppose that  $\mathcal{A}$  is an  $\mathcal{L}$ -structure and  $s$  is a variable assignment function into  $\mathcal{A}$ . The function  $\bar{s}$ , called the **term assignment function generated by  $s$** , is the function with domain consisting of the set of  $\mathcal{L}$ -terms and codomain  $A$  defined recursively as follows:

1. if  $t$  is a variable,  $\bar{s}(t) = s(t)$ .
2. if  $t$  is a constant symbol  $c$ ,  $\bar{s}(t) = c^{\mathcal{A}}$ .
3. if  $t \equiv f t_1 t_2 \dots t_n$  then  $\bar{s}(t) = f^{\mathcal{A}}(\bar{s}(t_1), \bar{s}(t_2), \dots, \bar{s}(t_n))$

Now we will describe truth (or satisfaction) for arbitrary formulas, relative to an assignment function.

**Definition 10** –Suppose that  $\mathcal{A}$  is an  $\mathcal{L}$ -structure,  $\phi$  is an  $\mathcal{L}$ -formula, and  $s : Vars \rightarrow A$  is an assignment function. We will say that  $\mathcal{A}$  satisfies  $\phi$  with assignments, and write  $\mathcal{A} \models \phi[s]$  in the following circumstances:

1. If  $\phi \equiv t_1 t_2$  and  $\bar{s}(t_1)$  is the same element of the universe  $A$  as  $\bar{s}(t_2)$ , or
2. If  $\phi \equiv R t_1 \dots t_n$  and  $(\bar{s}(t_1) \dots \bar{s}(t_n)) \in R^{\mathcal{A}}$ , or
3. If  $\phi \equiv (\neg \alpha)$  and  $\mathcal{A} \not\models \alpha[s]$ , or
4. If  $\phi \equiv (\alpha \vee \beta)$  and  $\mathcal{A} \models \alpha[s]$  or  $\mathcal{A} \models \beta[s]$  (or both), or
5. If  $\phi \equiv (\forall x)(\alpha)$  and for each  $a \in A$ ,  $\mathcal{A} \models \alpha[s[x|a]]$ .

And if we have a set of formulas  $\Gamma$ , we say that  $\mathcal{A} \models \Gamma[s]$  if for each formula  $\gamma$  in  $\Gamma$ ,  $\mathcal{A} \models \gamma$ . Now to talk about truth or falsity of a sentence in a structure, we will take our definition of satisfaction relative to an assignment function and prove that for sentences, the choice of the assignment function is inconsequential. Then we'll have shown that a sentence is true in a structure if and only if for any (therefore for all) choice of assignment function it is satisfied in structure. Here we need a lemma and a proposition but the proof of which is left to the reader + The context is borrowed from the book *Friendly Introduction to Mathematical Logic* and therefore one can check the page 30 for proofs.

**Lemma 1** –Suppose that  $s_1$  and  $s_2$  are variable assignment functions into a structure  $\mathcal{A}$  such that  $s_1(v) = s_2(v)$  for every variable  $v$  in the term  $t$ . Then  $\bar{s}_1(t) = \bar{s}_2(t)$ .

**Proposition 1** –Suppose that  $s_1$  and  $s_2$  are variable assignment functions into a structure  $\mathcal{A}$  such that  $s_1(v) = s_2(v)$  for every free variable  $v$  in the formula  $\phi$ . Then  $\mathcal{A} \models \phi[s_1]$  if and only if  $\mathcal{A} \models \phi[s_2]$ .

**Corollary 1** –If  $\sigma$  is a sentence in the language  $\mathcal{L}$  and  $\mathcal{A}$  is a structure of the language, either  $\mathcal{A} \models \sigma[s]$  for all assignment functions or for no assignment function.

And lastly we would define models:

**Definition 11** –If  $\phi$  is a formula in the language  $\mathcal{L}$  and  $\mathcal{A}$  is an  $\mathcal{L}$ -structure, we say that  $\mathcal{A}$  is a model of  $\phi$ , and write  $\mathcal{A} \models \phi$  if and only if  $\mathcal{A} \models \phi[s]$  for every assignment function  $s$ .

Notice that if  $\sigma$  is a sentence, then  $\mathcal{A} \models \sigma$  if and only if it is so in every possible assignment function  $s$ . In such a case we would say that the sentence is true in  $\mathcal{A}$ .

## Substitution and Substitutability

**§ Substitution:** Suppose that  $\mathcal{A} \models \forall x \exists y \neg(x = y)$ . This sentence is, in fact, true in any structure  $\mathcal{A}$  such that  $A$  has at least two elements. If we then proceed to replace the variable  $x$  by variable  $u$ , we get the statement  $\exists y \neg(u = y)$ , which will still be true in  $\mathcal{A}$ , no matter what value we give to the variable  $u$ . However if we take the formula and replace  $x$  by  $y$  then we got:  $\exists y \neg(y = y)$ , which will be false in any structure. So by a poor choice of substituting variable, we have changed the truth values of our formula. The rules of substitutability that we will discuss in this section are designed to help us avoid this problem, the problem of attempting to substitute a term inside a quantifier that binds a variable that involved in the term.

**Definition 12** –Suppose that  $u$  is a term,  $x$  is a variable and  $t$  is a term in language  $\mathcal{L}$ . We define the term  $u_t^x$  (read  $u$  with  $x$  replaced by  $t$ ) as follows:

1. If  $u$  is a variable not equal to  $x$  then  $u_t^x$  is  $u$ ,
2. If  $u$  is  $x$  then  $u_t^x$  is  $t$ ,
3. If  $u$  is a constant symbol, then  $u_t^x$  is  $u$ ,
4. If  $u \equiv f u_1 u_2 \dots u_n$  then  $u_t^x \equiv f(u_1)_t^x (u_2)_t^x \dots (u_n)_t^x$ .

And also for formulas:

**Definition 13** –Suppose that  $\phi$  is a formula in the language,  $t$  is a term and  $x$  is a variable. We define formula  $\phi_t^x$  (read the formula  $\phi$  with  $x$  replaced by  $t$ ) as follows:

1. If  $\phi := u_1 u_2$  then  $\phi_t^x := (u_1)_t^x (u_2)_t^x$ .
2. If  $\phi := R u_1 \dots u_n$  then  $\phi_t^x := R (u_1)_t^x \dots (u_n)_t^x$ .
3. If  $\phi := \neg(\alpha)$  then  $\phi_t^x := \neg(\alpha_t^x)$ .
4. If  $\phi := (\alpha \vee \beta)$  then  $\phi_t^x$  is  $(\alpha_t^x \vee \beta_t^x)$ .
5. If  $\phi := (\forall y)(\alpha)$  then:

$$\phi_t^x = \begin{cases} \phi & x = y \\ (\forall y)(\alpha_t^x) & \text{otherwise} \end{cases}$$

Having defined what we mean when we substitute a term for a variable, we will now define what it means for a term to be substitutable for a variable in a formula. The idea is that if  $t$  is substitutable for  $x$  in  $\phi$ , we will not run into the problems discussed at the beginning of this section. We will not substitute a term in such a way that a variable contained in that term is inadvertently bound by a quantifier.

**Definition 14** –Suppose  $\phi$  is an  $\mathcal{L}$ -formula,  $t$  is a term, and  $x$  is a variable. We say that  $t$  is substitutable for  $x$  in  $\phi$  if:

1.  $\phi$  is atomic, or
2.  $\phi := \neg(\alpha)$  and  $t$  is substitutable in  $\alpha$ , or
3.  $\phi := (\alpha \vee \beta)$  and  $t$  is substitutable for  $x$  in both  $\alpha$  and  $\beta$ , or
4.  $\phi := (\forall y)(\alpha)$  and either  $x$  is not free in  $\phi$  or  $y$  does not occur in  $t$  and  $t$  is substitutable for  $x$  in  $\alpha$ .

**§ Logical Implication:** At first glance it seems that a large portion of mathematics can be broken down into answering questions of the form: If you know such statement is true, is it the case that this other statement is also true? In this point we would formalize this question:

**Definition 15** –Suppose that  $\Delta$  and  $\Gamma$  are sets of  $\mathcal{L}$ -formulas. We will say that  $\Delta$  logically implies  $\Gamma$  and write  $\Delta \models \Gamma$  if for every  $\mathcal{L}$ -structure  $\mathcal{A}$ , if  $\mathcal{A} \models \Delta$ , then  $\mathcal{A} \models \Gamma$ .

This definition is a little bit tricky. It says that if  $\Delta$  is true in  $\mathcal{A}$ , then  $\Gamma$  is true in  $\mathcal{A}$ .

**Definition 16** –An  $\mathcal{L}$  formula  $\phi$  is said to be valid if  $\emptyset \models \phi$ , in other words, if  $\phi$  is true in every structure of language with every assignment function  $s$ . In this case, we will write  $\models \phi$ .