MATH 150 FALL 2012

PREDICATE LOGIC: SYNTAX AND SEMANTICS

1. Language of predicate logic $\mathcal{L}$. Symbols of the language:

(i) Logical symbols:
   (a) Infinitely many variables $v_0, v_1, v_2, \dots$.
   (b) Symbol for equality $\doteq$.
   (c) Sentential connectives $\neg$ and $\wedge$.
   (d) Existential quantifier $\exists$.
   (e) Parentheses (and).

(ii) Special symbols:
   (a) Constant symbols $c, d, e, \dots$.
   (b) Function symbols $F, G, H, \dots$. To each function symbol we assign an integer $n > 0$ called **arity** of the symbol. The arity of function symbol F expresses the number of arguments. If the arity of F is n we say that F is an n**-ary** or n**-place** function symbol.
   (c) Relational symbols $R, S, T, \dots$. To each relational symbol we assign an integer $n > 0$ called **arity** of the symbol. The arity of relational symbol R expresses the number of arguments. If the arity of R is n we say that R is an n**-ary** or n**-place** relational symbol.

Conventions:

(a) For variables often write $u, v, w, \dots$ in place of $v_0, v_1, v_2, \dots$.
(b) The remaining sentential connectives $\vee, \to$ and $\leftrightarrow$ are considered abbreviations $\neg(\neg - \wedge \neg -)$, $\neg(- \wedge \neg -)$, and $(- \to -) \wedge (- \to -)$, respectively.
(c) The universal quantifier $\forall$ is considered an abbreviation for $\neg\exists\neg$.
(d) To specify the language $\mathcal{L}$ it amounts to specify the special symbols of $\mathcal{L}$. We do this by listing them in the following form: We write $\mathcal{L} = \{c, d, e, \dots, F, G, H, \dots, R, S, T, \dots\}$ and say which of them are constant, function and relational symbols. Also, for function and relational symbols we specify the arity.

2. Structures for languages. Given a language $\mathcal{L}$, a structure $\mathcal{M}$ for $\mathcal{L}$ is given by the following data:

- A set M called the **domain** of structure $\mathcal{M}$.
- For each constant symbol c an object $c^{\mathcal{M}} \in M$.
- For each n-place function symbol F a function $F^{\mathcal{M}} : \underbrace{M \times \cdots \times M}_{n\text{-times}} \to M$.
- For each n-place relational symbol R a relation $R^{\mathcal{M}} \subseteq \underbrace{M \times \cdots \times M}_{n\text{-times}}$.

The object $c^{\mathcal{M}}$, resp. function $F^{\mathcal{M}}$, resp. relation $R^{\mathcal{M}}$ is called the **interpretation** of the constant symbol c, resp. of the function symbol F, resp. of the relational symbol R in the structure $\mathcal{M}$.

3. Terms. Terms are strings of $\mathcal{L}$-symbols defined by induction on complexity as follows.

(a) The one element string

$$c$$

where c is a constant symbol, as well as the one element string

$$v$$

where v is a variable, is a term. These one element terms are called **atomic** terms.

(b) If F is an n-place function symbol and $t_1, \ldots, t_n$ are terms then the string

$$F(t_1 t_2 \ldots t_n)$$

is a term. We informally write $F(t_1, t_2, \ldots, t_n)$ to make the term easier legible, but we do not consider the commas here a part of our language.

(c) Nothing else is a term.

4. Evaluations of variables. Given an $\mathcal{L}$-structure $\mathcal{M}$ with domain M, an **evaluation of variables** in $\mathcal{M}$ is a function

$$s : V \to M$$

where V, the domain of the function s, is a finite set of variables. In other words, s is an evaluation of variables in $\mathcal{M}$ if s is a function whose domain is a finite set of variables, and which assigns to each variable from its domain an element of M. We write $\mathrm{dom}(s)$ to denote the domain of s.

5. Evaluations of terms. Given a language $\mathcal{L}$, an $\mathcal{L}$-structure $\mathcal{M}$, a term t in this language (we also say "$\mathcal{L}$-term"), and an evaluation of variables s in $\mathcal{M}$, we define the **evaluation of t in $\mathcal{M}$, denoted by $t^{\mathcal{M}}[s]$,** by induction on complexity of t:

(a) If t is the one-element string c where c is a constant symbol then

$$t^{\mathcal{M}}[s] = c^{\mathcal{M}}.$$

(b) If t is the one-element string v where v is a variable then

$$t^{\mathcal{M}}[s] = s(v),$$

granting that $v \in \mathrm{dom}(s)$. If $v \notin \mathrm{dom}(s)$ then $t^{\mathcal{M}}[s]$ is undefined.

(a) If t is of the form $F(t_1 \ldots t_n)$ where F is an n-place function symbol and $t_1, \ldots, t_n$ are terms then

$$t^{\mathcal{M}}[s] = F^{\mathcal{M}}(t_1^{\mathcal{M}}[s], \ldots, t_n^{\mathcal{M}}[s]),$$

granting that $t_1^{\mathcal{M}}[s], \ldots, t_n^{\mathcal{M}}[s]$ are defined. Otherwise $t^{\mathcal{M}}[s]$ is undefined.

It is intuitively clear that $t^{\mathcal{M}}[s]$ depends only on the values of s on the variables that occur in t.

Conventions.

A. We often write $t(v_1, \ldots, v_n)$ to display the variables we are interested in. In this notation, if s is an evaluation such that $v_1, \ldots, v_n \in \mathrm{dom}(s)$ and $s(v_1) = a_1, \ldots, s(v_n) = a_n$ we write $t^{\mathcal{M}}(v_1, \ldots, v_n)[a_1, \ldots, a_n]$ or simply $t^{\mathcal{M}}[a_1, \ldots, a_n]$ in place of $t^{\mathcal{M}}[s]$. Since we are primarily interested in the situations where $t^{\mathcal{M}}[s]$ is defined, typically we assume that all variables occurring in t are among the variables $v_1, \ldots, v_n$. (Although the official definition does not require this.)

6. Formulas. Given a language $\mathcal{L}$ we define the notion of $\mathcal{L}$-formula by induction on complexity.

(a) Base step.

(i) If t, t' are $\mathcal{L}$-terms then the string

$$t \dot{=} t'$$

is an $\mathcal{L}$-formula.

(ii) If R is an n-place relational symbol of $\mathcal{L}$ and $t_1, \dots, t_n$ are $\mathcal{L}$-terms then the string

$$R(t_1 \dots t_n)$$

is an $\mathcal{L}$-formula.

Formulas defined in (a) are called **atomic** formulas.

(b) Induction step.

(i) If φ is an $\mathcal{L}$-formula then the string

$$\neg\varphi$$

is also an $\mathcal{L}$-formula.

(ii) If φ and φ' are $\mathcal{L}$-formulas then also the string

$$(\varphi \wedge \varphi')$$

is an $\mathcal{L}$-formula.

(iii) If φ is an $\mathcal{L}$-formula and v is a variable then the string

$$(\exists v)\varphi$$

is an $\mathcal{L}$-formula.

(c) Nothing else is an $\mathcal{L}$-formula.

Conventions:

A. If the language $\mathcal{L}$ is clear from the context we say briefly "formula" instead of "$\mathcal{L}$-formula".

B. If φ is a formula and $v_1, \dots, v_\ell$ are variables we often write "$\varphi(v_1, \dots, v_\ell)$" to display the variables we are interested in. This expression is however informal, and our official definition should be kept in mind whenever doing rigorous arguments involving formulas. The above expression is used to make the text easier legible. The variables $v_1, \dots, v_\ell$ typically come from the formula φ, but also may not: Although this may seem to make no use/sense, it is a convenient notation.

C. The remaining connectives are considered mere abbreviations. Thus, the string $(\varphi \vee \varphi')$ is considered an abbreviation for $\neg(\neg\varphi \wedge \neg\varphi')$, and similarly we define the meaning $(\varphi \to \varphi')$ and $(\varphi \leftrightarrow \varphi')$ in terms of $\neg$ and $\wedge$. Also, the universal quantification $(\forall v)\varphi$ is considered an abbreviation for $\neg(\exists v)\neg\varphi$.

7. Semantics: The satisfaction relation $\models$. Given a language $\mathcal{L}$, an $\mathcal{L}$-structure $\mathcal{M} = (M, c, d, e, \dots, F, G, H, \dots, R, S, T, \dots)$ (so the domain of the structure $\mathcal{M}$ is M), an $\mathcal{L}$-formula φ, and an evaluation s of variables in $\mathcal{M}$, we define the notion

$$\mathcal{M} \models \varphi[s],$$

which we read as "the structure $\mathcal{M}$ satisfies the formula φ at the evaluation s", or "the formula φ holds in $\mathcal{M}$ at the evaluation s", by induction on complexity of φ.

(a) Base step: Atomic formulas.

(i) If φ is of the form

$$t \doteq t'$$

where t, t' are terms, we define

$$\mathcal{M} \models \varphi[s]$$

if and only if

$$t^{\mathcal{M}} = t'^{\mathcal{M}},$$

granting that the evaluations of terms $t^{\mathcal{M}}, t'^{\mathcal{M}}$ are defined. Otherwise we leave "$\mathcal{M} \models \varphi[s]$" undefined.

(ii) If φ is of the form

$$R(t_1 \dots t_n)$$

where R is an n-place relational symbol and $t_1, \dots, t_n$ are terms, we define

$$\mathcal{M} \models \varphi[s]$$

if and only if

$$(t_1^{\mathcal{M}}[s], \dots, t_n^{\mathcal{M}}[s]) \in R^{\mathcal{M}},$$

granting that the evaluations of terms $t_1^{\mathcal{M}}, \dots, t_n^{\mathcal{M}}$ are defined. Otherwise we leave "$\mathcal{M} \models \varphi[s]$" undefined.

(b) Induction step.

(i) If φ is of the form

$$\neg\psi$$

and the meaning of "$\mathcal{M} \models \psi[s]$" is already defined for all s, we let

$$\mathcal{M} \models \varphi[s] \quad \text{if and only if} \quad \mathcal{M} \not\models \psi[s].$$

(ii) If φ is of the form

$$(\psi \wedge \psi')$$

and the meaning of "$\mathcal{M} \models \psi[s]$" and "$\mathcal{M} \models \psi'[s]$" is already defined for all s, we let

$$\mathcal{M} \models \varphi[s] \qquad \text{if and only if} \qquad \mathcal{M} \models \psi[s] \text{ and } \mathcal{M} \models \psi'[s].$$

(iii) If φ is of the form

$$(\exists v)\psi$$

and the meaning of "$\mathcal{M} \models \psi[s']$" is already defined for all s', we let

$$\mathcal{M} \models \varphi[s]$$

if and only if there is an evaluation of variables s' satisfying all of the following:

(1) $\mathrm{dom}(s') = \mathrm{dom}(s) \cup \{v\}$.
(2) $s'(u) = s(u)$ for all $u \in \mathrm{dom}(s)$.
(3) $\mathcal{M} \models \psi[s']$.

Conventions.

A. If we display variables in φ, that is, if we write informally $\varphi(v_1, \ldots, v_\ell)$, and if s is an evaluation of variables in $\mathcal{M}$ such that $s(v_1) = a_1, \ldots, s(v_\ell) = a_\ell$ then we often write informally

$$\mathcal{M} \models \varphi(v_1, \ldots, v_n)[a_1, \ldots, a_\ell],$$

or even more casually

$$\mathcal{M} \models \varphi[a_1, \ldots, a_\ell]$$

in place of

$$\mathcal{M} \models \varphi[s]$$

In the latter case, we assume that it is clear from the context what variables are in φ and how they are evaluated by s.

B. Using Convention A we rewrite the definition of satisfaction in the following "casual", but easier-to-read way. See Convention 5A for notation concerning terms.

(a) Atomic formulas.

(i) $\mathcal{M} \models (t \dot{=} t')(v_1, \ldots, v_n)[a_1, \ldots, a_n]$ iff

$$t^{\mathcal{M}}(v_1, \ldots, v_n)[a_1, \ldots, a_n] = t'^{\mathcal{M}}(v_1, \ldots, v_n)[a_1, \ldots, a_n].$$

(ii) If R is a k-place relational symbol and $t_1, \ldots, t_k$ are terms then

$$\mathcal{M} \models R(t_1 \ldots t_k)(v_1, \ldots, v_n)[a_1, \ldots, a_n]$$

iff

$$(t_1^{\mathcal{M}}(v_1, \ldots, v_n)[a_1, \ldots, a_n], \ldots, t_k^{\mathcal{M}}(v_1, \ldots, v_n)[a_1, \ldots, a_n]) \in R^{\mathcal{M}}.$$

(b) Induction step.

(i) $\mathcal{M} \models \neg\varphi(v_1, \ldots, v_n)[a_1, \ldots, a_n)$ iff

$$\mathcal{M} \not\models \varphi(v_1, \ldots, v_n)[a_1, \ldots, a_n).$$

(ii) $\mathcal{M} \models (\varphi \wedge \varphi')(v_1, \ldots, v_n)[a_1, \ldots, a_n)$ iff

$$\mathcal{M} \models \varphi(v_1, \ldots, v_n)[a_1, \ldots, a_n) \text{ and } \mathcal{M} \models \varphi'(v_1, \ldots, v_n)[a_1, \ldots, a_n).$$

(iii) $\mathcal{M} \models (\exists v)\varphi(v, v_1, \ldots, v_n)[a_1, \ldots, a_n)$ iff there is some $b \in M$ such that

$$\mathcal{M} \models \varphi(v, v_1, \ldots, v_n)[b, a_1, \ldots, a_n).$$

Again, following Convention A we will be even more casual, and drop displaying the variables $v_1, \ldots, v_n$, assuming they are clear from the context. So for instance (b)(iii) above will be written as: $\mathcal{M} \models (\exists v)\varphi[a_1, \ldots, a_n]$ iff there is some $b \in M$ such that $\mathcal{M} \models \varphi[b, a_1, \ldots, a_n]$. Here it is assumed that it is clear from the context what variables are in φ and how they are evaluated.

8. Free and bound variables. Recall that a formula φ is a string, that is, a finite sequence of symbols. So we can index this sequence with nonnegative integers and write

$$\varphi = (z_0, z_1, \ldots, z_{\ell-1});$$

here ℓ is the **length** of the string φ often denoted by $\mathsf{lh}(\varphi)$. We say that a number $i < \mathsf{lh}(\varphi)$ is an **occurrence** of the symbol z in φ if and only if $z_i = z$.

Informally, an occurrence of a variable v in a formula φ is called **free** if and only this occurrence is not under the influence of any quantifier in φ. Otherwise the occurrence is **bound**. So for instance if φ is the formula

$$F(uv)\doteq w \wedge (\exists v)F(uv)\doteq F(uu)$$

then $\mathsf{lh}(\varphi) = 23$, the first occurrence of variable v in φ is 3 (we count from 0), the second occurrence of v in φ is 11, and the third one is 16. The occurrence 3 is free, the occurrences 11 and 16 are bound.

Inductively we define the free occurrence of a variable v in the formula φ as follows.

Definition. Fix a language $\mathcal{L}$.

(a) Atomic formulas. If φ is an atomic formula then any occurrence of v in φ is free.

(b) Induction step.

  (i) If φ is of the form $\neg\psi$ then an occurrence i of v in φ is free if and only if the occurrence $i-1$ of v in ψ is free.

  (ii) If φ is of the form $(\psi \wedge \psi')$ then an occurrence i of v in φ is free if and only if one of the following holds.

    (A) $i < \mathsf{lh}(\psi) + 1$ and $i - 1$ is a free occurrence of v in ψ.

    (B) $i > \mathsf{lh}(\psi) + 1$ and $i - \mathsf{lh}(\psi) - 2$ is a free occurrence of v in ψ'.

  (iii) If φ is of the form $(\exists u)\psi$ then i is a free occurrence of v in φ if both the following hold.

    (A) $i - 4$ is a free occurrence of v in ψ.

    (B) v is a variable distinct from u.

Write down some formulas and see how this definition works in practice.

9. Free variables and satisfaction relation. The truth of a formula depends only on the evaluation of its free variables. The following proposition makes this precise. Please formulate Proposition 1 in the lecture, but do not go into the proof of the proposition – the proof should be an exercise done during the discussion. Part of it will be a homework assignment.

Proposition 1. If φ is a formula and s is an evaluation of variables in a structure $\mathcal{M}$ then

(a) "$\mathcal{M} \models \varphi[s]$" is defined if and only if s evaluates all variables with free occurrence in φ, that is, each such variable is in $\mathrm{dom}(s)$.

(b) "$\mathcal{M} \models \varphi[s]$" depends only on the evaluation of variables with free occurrence in φ. That is, if s, s' are evaluations of variables in $\mathcal{M}$ and $s(v) = s'(v)$ whenever v has free occurrence in φ then

$$\mathcal{M} \models \varphi[s] \quad \text{if and only if} \quad \mathcal{M} \models \varphi[s']$$

Proposition 1 is proved by induction on complexity of φ.

In view of Proposition 1 we can give now a more precise meaning to the notation "$\varphi(v_1, \ldots, v_n)$": when we write this, we are primarily interested in the case where $v_1, \ldots, v_n$ are variables with free occurrence in φ. Similarly, when we write $\mathcal{M} \models \varphi[a_1, \ldots, a_n]$ then we are primarily interested in the case where $a_1, \ldots, a_n$ evaluate variables $v_1, \ldots, v_n$ with free occurrence in φ.

Definition. A formula φ is a **sentence** if and only if all occurrences of all variables in φ are bound.

By Proposition 1, if φ has variables with free occurrences then the truth value of φ in the structure $\mathcal{M}$ depends on how we evaluate such variables. Now if φ is a sentence, its truth value in $\mathcal{M}$ does not depend on any evaluation of variables, i.e. it is either always true or always false. Sentences are formal variants of mathematical theorems, as they can be assigned truth value without adding any further more specific information. So for instance "there are infinitely many primes" is a meaningful statement which can be translated into a sentence in the formal language of arithmetic.

Definition. Given is a language $\mathcal{L}$ and an $\mathcal{L}$-structure $\mathcal{M}$.

(a) An $\mathcal{L}$-sentence σ **holds/is true** in the $\mathcal{M}$ if and only if $\mathcal{M} \models \sigma[s]$ for some/all evaluation of variables s. By Proposition 1 and the discussion following this proposition, this statement is independent of s, so it is OK to say some/all. We will write briefly

$$\mathcal{M} \models \sigma.$$

(b) A set of $\mathcal{L}$-sentences Σ **holds/is true** in $\mathcal{M}$ if and only if $\mathcal{M} \models \sigma$ for every $\sigma \in \Sigma$. We write

$$\mathcal{M} \models \Sigma.$$

Definition. A set of $\mathcal{L}$-sentences Σ is **satisfiable** if and only if there is an $\mathcal{L}$-structure $\mathcal{M}$ such that $\mathcal{M} \models \Sigma$. A satisfiable set of sentences is also called a **theory**. If we want to specify the language, we talk about **$\mathcal{L}$-theory**. Refer to the link "THEORIES" on the course webpage for some important theories we will be considering.

Definition. (Model of a theory) Consider a language $\mathcal{L}$. Let Σ be an $\mathcal{L}$-theory, that is, a satisfiable set of $\mathcal{L}$-sentences. An $\mathcal{L}$-structure $\mathcal{M}$ is called a **model of** Σ if and only if $\mathcal{M} \models \Sigma$.

Definition. (Theory of a structure) Consider a language $\mathcal{L}$ and an $\mathcal{L}$-structure $\mathcal{M}$. The **complete theory of** $\mathcal{M}$ is the set of $\mathcal{L}$-sentences

$$\mathbf{Th}_{\mathcal{L}}(\mathcal{M}) = \{\sigma \mid \sigma \text{ is an } \mathcal{L}\text{-sentence and } \mathcal{M} \models \sigma\}.$$

Thus, $\mathbf{Th}_{\mathcal{L}}(\mathcal{M})$ consists of all "true statements" about the model that can be expressed in the language $\mathcal{L}$. If the language $\mathcal{L}$ is clear from the context, we write $\mathbf{Th}(\mathcal{M})$ in place of $\mathbf{Th}_{\mathcal{L}}(\mathcal{M})$.

Definition. (Logical/Semantical implication) Consider a language $\mathcal{L}$ and a set Σ of $\mathcal{L}$-sentences. Let σ be an $\mathcal{L}$ sentence. We say that Σ **logically** or **semantically implies** σ, and write

$$\Sigma \models \sigma$$

if and only if for every $\mathcal{L}$-structure $\mathcal{M}$.

$$\text{If } \mathcal{M} \models \Sigma \quad \text{then} \quad \mathcal{M} \models \sigma.$$

In other words, $\Sigma \models \sigma$ if and only if $\mathcal{M} \models \sigma$ whenever $\mathcal{M}$ is a model of Σ. If τ is an $\mathcal{L}$-sentence, we write briefly $\tau \models \sigma$ in place of rigorous $\{\tau\} \models \sigma$.

More generally we define the logical/semantical implication for arbitrary set of formulas $\Sigma(v_1, \ldots, v_\ell)$ and a formula $\sigma(v_1 \ldots, v_\ell)$, where all free variables in formulas in Σ and in σ are among $v_1, \ldots, v_\ell$, as follows:

$$\Sigma \models \sigma$$

if and only if for every $\mathcal{L}$-structure $\mathcal{M}$ and every evaluation of variables s in $\mathcal{M}$,

$$\text{If } \mathcal{M} \models \tau[s] \text{ for every } \tau \in \Sigma \quad \text{then} \quad \mathcal{M} \models \sigma[s].$$

Notice that the symbol $\models$ is used in two different meanings: For the satisfaction relation, and also for logical implication. It is always clear from the context which use we mean. Do not confuse the two!

10. Definable sets. Sets definable in an $\mathcal{L}$-structure $\mathcal{M}$ are those sets which can be described by some $\mathcal{L}$-formula. The following definition makes this precise.

Definition. Given an $\mathcal{L}$-structure $\mathcal{M}$ with domain M, and a set

$$A \subseteq \underbrace{M \times \cdots \times M}_{n-\text{times}},$$

we say that A is **definable** in the structure $\mathcal{M}$ if and only if there is an $\mathcal{L}$-formula $\varphi(u_1, \ldots, u_n, v_1, \ldots, v_\ell)$ and some $a_1, \ldots, a_\ell \in M$ such that

$$A = \{(x_1, \ldots, x_n) \in \underbrace{M \times \cdots \times M}_{n-\text{times}} \mid \mathcal{M} \models \varphi[x_1, \ldots, x_n, a_1, \ldots, a_\ell]\}.$$

If A is definable in $\mathcal{M}$ without parameters we also say that A is **lightface** definable in $\mathcal{M}$; if A is definable with parameters then we say that A is **boldface** definable in $\mathcal{M}$. The formula φ is then called a **definition** of A in $\mathcal{M}$.

The same set A may have many different definitions in $\mathcal{M}$, some of them may be with parameters, and some of them may be without parameters. So lightface definability is a more specific notion than boldface definability.

11. Substitution and substitutability. Given a formula φ, a variable v and a term t, we define the following:

(a) The result of **substitution** of t for v in φ, denoted by $\varphi(v/t)$. Here $\varphi(v/t)$ is defined to be the formula obtained by substituting t for every **free** occurrence of v in φ, granting that t is **substitutable** for v in φ.

(b) The **substitutability** of t for v in φ. We say that t **substitutable** for v in φ if and only if no occurrence of a variable u in t results in a bounded occurrence of u after substituting t for v in φ.

These notions are tricky, but make intuitive sense. We substitute only for free occurrences, as the meaning of bound occurrences is given by the quantifiers. So it would not make reasonable sense to substitute for bounded occurrences. However, even if we substitute for free occurrences, there are some issues – the result may make sense, but may change the meaning of the formula. Here is an example. Let $\mathcal{L}$ be a language of Arithmetic. Consider the following formula:

$$\varphi(v) \equiv (\exists u)(v \dot{=} u \dot{\times} u).$$

Obviously the variable v has only one occurrence in φ, and this occurrence is free. Now consider the following terms:

$$t \equiv w\dot{+}w' \quad t' \equiv w\dot{+}v \quad t'' \equiv w\dot{+}u.$$

The results of the substitution of these terms for v in φ are:

$$\begin{array}{rcl} \varphi(v/t) & \equiv & (\exists u)(w\dot{+}w'\dot{=}u\dot{\times}u) \\ \varphi(v/t') & \equiv & (\exists u)(w\dot{+}v\dot{=}u\dot{\times}u) \\ \varphi(v/t'') & \equiv & (\exists u)(w\dot{+}u\dot{=}u\dot{\times}u) \end{array}$$

Notice the difference between $\varphi(v,t), \varphi(v/t')$ on the one side, and $\varphi(v/t'')$ on the other side: The formulas $\varphi(v/t), \varphi(v/t')$ have meaning similar to that of φ – after evaluating in the structure $\mathbb{N}$, they demand that certain term which does not contain u as a variable receives value $a \cdot a$ where u is evaluated by $a \in N$. The formula $\varphi(v/t'')$ demands more, namely it describes the difference $a \cdot a - a$. Notice also that in the case of substituting t'', the occurrence of u in t'' becomes bound in $\varphi(v/t'')$. For this reason – that is – for the reason that this kind of substitution changes the meaning of the formula, we disallow this kind of substitution, and this is precisely what the notion of substitutability does.

Both notions of substitutability and substitution can be defined inductively, but we first have to define the notion of substitutability in order to disallow unwanted kind of substitution.

Inductive definition of Substitutability. Consider a formula φ, a variable v and a term t.

(a) If φ is an atomic formula then t is substitutable for v in φ.
(b) If φ is of the form $\neg\psi$ then t is substitutable for v in φ if and only if t is substitutable for v in ψ.
(c) If φ is of the form $(\psi \wedge \psi')$ then t is substitutable for v in φ if and only if t is substitutable for v in both ψ and ψ'.
(d) If φ is of the form $(\exists u)\psi$ then t is substitutable for v in φ if and only if both the following are satisfied:
   (i) t is substitutable for v in ψ.
   (ii) Either v has no free occurrence in φ or else u has no occurrence in t.

It follows from the rigorous inductive definition of $\varphi(v/t)$ below that in (d)(iii) the case "v has no free occurrence in φ" is the trivial case where we will not actually perform any substitution. We first define the substitution $t'(v/t)$ of t for v in a term t'.

Inductive definition of $t'(v/t)$. Let t, t' be terms and v be a variable.

(a) If t' is the string c consisting of a single constant symbol c then $t'(v/t)$ is the term t'.
(b) If t' is the string u consisting of a single variable u then $t'(v/t)$ is the term t' if the variables u, v are distinct, and $t'(v/t)$ is the term t if u is the same variable as v.
(c) If t' is the term $F(t_1 \ldots t_\ell)$ where F is an ℓ-place function symbol and $t_1, \ldots, t_\ell$ are terms then $t'(v/t)$ is the term $F(t_1(v/t) \ldots t_\ell(v/t))$.

Write down examples, say in the language of Arithmetic, starting from the simplest ones and look at what is happening!

Inductive definition of $\varphi(v/t)$. Let φ be a formula, v be a variable, and t be a term.

(a) If φ is an atomic formula then $\varphi(v/t)$ is defined as follows.
  (i) If φ is of the form $t_1 \doteq t_2$ where t_1, t_2 are terms, then $\varphi(v/t)$ is the formula $t_1(v/t) \doteq t_2(v/t)$.
  (ii) If φ is of the form $R(t_1 \dots t_\ell)$ where R is an ℓ-place relational symbol and $t_1 \dots t_\ell$ are terms then $\varphi(v/t)$ is the formula $R(t_1(v/t) \dots r_\ell(v/t))$.
(b) If φ is of the form $\neg\psi$ then $\varphi(v/t)$ is the formula $\neg\psi(v/t)$.
(c) If φ is of the form $(\psi \wedge \psi')$ then $\varphi(v/t)$ is the formula $(\psi(v/t) \wedge \psi'(v/t))$.
(d) If φ is of the form $(\exists u)\psi$ and t is substitutable for v in φ then:
  (i) If v has no free occurrence in φ then $\varphi(v/t)$ is the formula φ.
  (ii) If v has a free occurrence in φ then $\varphi(v/t)$ is the formula $(\exists u)\psi(v/t)$.
(e) In all other cases is $\varphi(v/t)$ undefined.

As with the terms, write down examples, starting with the simplest ones, to see how this inductive definition works!

12. Tautology Given is a language $\mathcal{L}$. We say that an $\mathcal{L}$-formula φ is a **$\mathcal{L}$-tautology** (or briefly **tautology**, if $\mathcal{L}$ is clear from the context) if and only if there are

- A tautology φ^* in sentential logic in some sentential letters $A_1, \dots, A_\ell$, and
- $\mathcal{L}$-formulas $\varphi_1, \dots, \varphi_\ell$

such that φ is obtained from φ^* by replacing each occurence of A_i in φ^* by φ_i for all $i = 1, \dots, \ell$.

Example: Say φ^* is the following tautology in sentential logic in sentential letters A, B, C:

$$\varphi : (A \to B) \to \Big((B \to C) \to (A \to C)\Big),$$

and φ_i, $i = 1, 2, 3$ are as follows:

$$\begin{aligned} \varphi_1 : &\quad (\exists u)\psi \\ \varphi_2 : &\quad (\rho \wedge \sigma) \\ \varphi_3 : &\quad (\forall w)(\psi \to \sigma). \end{aligned}$$

Then the formula

$$\big((\exists u)\psi \to (\rho \wedge \sigma)\big) \to \Big(\big((\rho \wedge \sigma) \to (\forall w)(\psi \to \sigma)\big) \to \big((\exists u)\psi \to (\forall w)(\psi \to \sigma)\big)\Big)$$

is a tautology.

13. Axioms of Predicate Logic. The following formulas are called Axioms of Predicate Logic.

(T)	ALL $\mathcal{L}$-TAUTOLOGIES	
(E1)	$t \doteq t$	whenever t is an $\mathcal{L}$-term
(E2)	$t \doteq t' \to (\varphi(v/t) \to \varphi(v/t'))$	whenever t, t' are $\mathcal{L}$-terms, v is a variable, and φ is an atomic $\mathcal{L}$-formula
(Q)	$\varphi(v/t) \to (\exists v)\varphi$	whenever t is an $\mathcal{L}$-term substitutable for variable v in $\mathcal{L}$-formula φ.

Here Axioms (E1) and (E2) are called the **Equality Axioms** and (Q) is called the **Quantifier Axiom**.

14. Rules of inference. Rules of inference tell us how to derive formula from given formulas. We have two rules of inference: Modus Ponens (MP) and the Quantifier Rule (QR).

(MP) $$\frac{\begin{array}{l}\varphi \to \psi \\ \varphi\end{array}}{\psi}$$ whenever φ, ψ are $\mathcal{L}$-formulas

(QR) $$\frac{\varphi \to \psi}{(\exists v)\varphi \to \psi}$$ whenever φ, ψ are $\mathcal{L}$-formulas and variable v has no free occurence in ψ.

15. Formal deduction. Given is a language $\mathcal{L}$ and a set of $\mathcal{L}$-sequences Σ. A sequence $\varphi_1, \dots, \varphi_n$ of $\mathcal{L}$-formulas is called a **formal deduction** or a **formal proof** in $\mathcal{L}$ from Σ if and only if each φ_i on this sequence satisfies one of the following.

- φ_i is an Axiom of Predicate Logic or $\varphi_i \in \Sigma$.
- φ_i follows by one of the rules of inference applied to some of the formulas $\varphi_1, \dots, \varphi_{i-1}$.

The formulas in Σ are called **assumptions**. If φ is an $\mathcal{L}$-formula then a **formal proof of** φ **from** Σ **in** $\mathcal{L}$ is a formal proof $\varphi_1, \dots, \varphi_n$ from Σ in $\mathcal{L}$ such that φ_n is φ. If there is a formal proof of φ from Σ in $\mathcal{L}$ we say that φ is **provable from** Σ **in** $\mathcal{L}$ and write

$$\Sigma \vdash_{\mathcal{L}} \varphi.$$

If the language $\mathcal{L}$ is clear from the context we write briefly $\Sigma \vdash \varphi$.

Here are some examples of formal proofs in an arbitrary language $\mathcal{L}$. Notice that here we do not use any particular set Σ of formulas that describe some concrete properties of $\mathcal{L}$-structures. On the left we label which Axiom or Rule of Inference is used. "(AS)" is the abbreviation for "assumption".

(AS)	φ
(Q)	$\varphi \to (\exists v)\varphi$
(MP)	$(\exists v)\varphi$

(AS)	$\neg(\exists v)\neg\varphi$
(Q)	$\neg\varphi \to (\exists v)\neg\varphi$
(T)	$\big(\neg\varphi \to (\exists v)\varphi\big) \to \big(\neg(\exists v)\neg\varphi \to \varphi\big)$
(MP)	$\neg(\exists v)\neg\varphi \to \varphi$
(MP)	φ

(AS)	φ
(T)	$\varphi \to \big(\neg\varphi \to \neg(\exists v)\varphi\big)$
(MP)	$\neg\varphi \to \neg(\exists v)\varphi$
(QR)	$(\exists v)\neg\varphi \to \neg(\exists v)\varphi$
(T)	$\big((\exists v)\neg\varphi \to \neg(\exists v)\varphi\big) \to \big((\exists v)\varphi \to \neg(\exists v)\neg\varphi\big)$
(MP)	$(\exists v)\varphi \to \neg(\exists v)\neg\varphi$
(Q)	$\varphi \to (\exists v)\varphi$
(MP)	$(\exists v)\varphi$
(MP)	$\neg(\exists v)\neg\varphi$

(AS)	$\varphi \to \psi$
(Q)	$\psi \to (\exists v)\psi$
(T)	$(\varphi \to \psi) \to \big(\psi \to (\exists v)\psi\big) \to \big(\varphi \to (\exists v)\psi\big)$
(MP)	$\big(\psi \to (\exists v)\psi\big) \to \big(\varphi \to (\exists v)\psi\big)$
(MP)	$\varphi \to (\exists v)\psi$
(QR)	$(\exists v)\varphi \to (\exists v)\psi$

The above examples give the formal proofs that demonstrate the following facts: $\{\varphi\} \vdash (\exists v)\varphi$, $\{(\forall v)\varphi\} \vdash \varphi$, $\{\varphi\} \vdash (\forall v)\varphi$ and $\{\varphi \to \psi\} \vdash (\exists v)\varphi \to (\exists v)\psi$.

16. Gödel's Completeness Theorem. Predicate logic is supposed to mimick our logical reasoning and reduce it to syntactical manipulations of formulas. In order that this indeed mimicks our reasoning, two things must hold: (A) whatever we can prove from true assumptions must be true, and (B) whatever is true can be proved. The property (A) is called Soundness; the property (B) is called Completeness. It can be proved that both hold. Here are the rigorous formulations.

Soundness Theorem. Assume $\mathcal{L}$ is a language, Σ is a set of $\mathcal{L}$-sentences, and σ is an $\mathcal{L}$-sentence. Then:

$$\text{If} \quad \Sigma \vdash \sigma \quad \text{then} \quad \Sigma \models \sigma.$$

Gödel's Completeness Theorem. Assume $\mathcal{L}$ is a language, Σ is a set of $\mathcal{L}$-sentences, and σ is an $\mathcal{L}$-sentence.

$$\text{If} \quad \Sigma \models \sigma \text{ then} \quad \Sigma \vdash \sigma.$$

The conjunction of the two theorems shows that the symbols $\vdash$ and $\models$ have equivalent meaning. But $\vdash$ is phrased in syntactical terms, whereas $\models$ is phrased in semantical terms.

We say that a set of sentences Σ is **consistent** if and only if it is not possible to prove a contradiction from Σ in $\mathcal{L}$. Precisely, there is no formula φ such that

$$\Sigma \vdash \varphi \quad \text{and} \quad \Sigma \vdash \neg\varphi.$$

Notice that if ψ is an arbitrary $\mathcal{L}$-formula then the following is a deduction:

(T)	$\varphi \to (\neg\varphi \to \psi)$
(AS)	φ
(MP)	$\neg\varphi \to \psi$
(AS)	$\neg\varphi$
(MP)	ψ

Thus, if both $\Sigma \vdash \varphi$ and $\Sigma \vdash \neg\varphi$ then we can put together the above formal proof together with the proofs of φ and $\neg\varphi$ from Σ in $\mathcal{L}$, and obtain a formal proof of ψ from Σ. This shows:

If Σ is not consistent then $\Sigma \vdash \psi$ for **every** $\mathcal{L}$-formula ψ.

Thus,

Σ is consistent <u>if and only if</u> there is an $\mathcal{L}$-formula ψ such that $\Sigma \nvdash \psi$.

We can now give equivalent formulations of Soundness and Completeness Theorems.

Soundness Theorem II. Assume $\mathcal{L}$ is a language and Σ is a set of $\mathcal{L}$-sentences. If Σ has a model, that is, if there is an $\mathcal{L}$-structure $\mathcal{M}$ such that $\mathcal{M} \models \Sigma$, then Σ is consistent.

Gödel's Completeness Theorem II. Assume $\mathcal{L}$ is a language and Σ is a set of $\mathcal{L}$-sentences. If Σ is consistent then Σ has a model, that is, there is an $\mathcal{L}$-structure $\mathcal{M}$ such that $\mathcal{M} \models \Sigma$.