Mathematical Logic

Hechen Hu

March 21, 2018

Contents

1 The Nature of Mathematical Logic **1**

1.1 Axiom Systems 1

1.2 Formal Systems 1

1.3 Syntactical Variables 3

2 First-Order Theories **5**

2.1 Functions and Predicates 5

2.2 Truth Functions 5

2.3 Variables and Quantifiers 6

2.4 First-Order Languages 6

2.5 Structures 8

2.6 Logical Axioms and Rules 10

1

The Nature of Mathematical Logic

1.1 Axiom Systems

Definition. *Axioms* are laws we accept without proof. *Theorems* are derived from axioms. *Basic concepts* are concepts not defined using other concepts, and *derived concepts* are defined in terms of these.

Definition. The entire edifice consists of basic concepts, derived concepts, axioms, and theorems is called an *axiom system*.

1.2 Formal Systems

Definition. Proofs which deal with concrete objects in a constructive manner are said to be *finitary*. For example, proof by contradiction is **not** finitary.

Definition. The study of axioms and theorems as sentences is called the *syntactical study* of axioms systems; the study of the meaning of these sentences is called the *semantical study* of axiom systems.

Definition. Any finite sequence of symbols of a language is called an *expression* of that language. In each language, certain expressions of the language are designated as the *formulas* of the language; it is intended that these be the expressions which assert some fact.

The language of a formal system F is denoted by $L(F)$.

Definition. A *rules of inference*, or *rules*, states that under certain conditions, one formula, called the *conclusion* of the rule, can be *inferred* from certain other formulas, called the *hypotheses* of the rule.

Definition. The theorems of a formal system F can be defined using a *generalized inductive definition*, that is

1. The axioms of F are theorems of F.
2. If all the hypotheses of a rule of F are theorems of F, then the conclusion of the rule is a theorem of F.

If we replace "are theorems of F" with "have property P", this is called *induction on theorems*, and it can be used to prove that every theorem of F has property P. The second part of our definition is called the *induction hypotheses.*

If a collection of objects is defined by a generalized inductive definition, the in order to prove that every object in it has property P, it suffices to prove that the objects having property P satisfy the laws of the definition.

Definition. A rule in formal system if *finite* if it has only finitely many hypotheses.

Definition. Let F be a formal system in which all the rules are finite. By a *proof* in F, we mean a finite sequence of formulas, each of which either is an axiom or is the conclusion of a rule whose hypotheses precede that formula in the proof. If A is the last formula in a proof P, we say that P is a proof of A.

Theorem 1.2.1. *A formula A is a theorem of F iff there is a proof of A.*

Proof. If A is an axiom of F, then certainly it has a proof. Now suppose that A can be inferred from $B_1, \cdots, B_n$ by some rule of F. By the induction hypothesis, each of the B_i has a proof. If we put these proofs one after the other, and add A to the end of this sequence, we obtain a proof for A. $\square$

We write $\vdash_F A$ as an abbreviation for A is a theorem of F. The subscript F can be omitted if no confusion results.

Definition. The symbols added to a language are called *defined symbols.* Each such symbol is to be combined in certain ways with symbols of the language and previously introduced defined symbols to form expressions called *defined formulas.* Each defined formula is to be an abbreviation of some formula of the language. With each defined symbol, one must give a *definition* of that symbol; that is a rule which tells how to form defined formulas with the new symbol and how to find, for each such defined formula, the formula of the given language which it abbreviates.

The defined symbols are **not** symbols of the language, and the defined formulas are **not** formulas of the language. When we say about a defined formula, we are talking about the formula of the language which it abbreviates. Thus the length of a defined formula is not the number of occurrences of symbols in the defined formula, but the number of occurrences of symbols in the formula which the defined formula abbreviates.

1.3 Syntactical Variables

Definition. *Syntactical variables* behave the same as variables in analysis. They vary through the expressions of the language being discussed and are fixed throughout one context. Syntactical variables are denoted with boldface letters. In particular, **u** and **v** would vary through all expressions, and **A**, **B**,**C**, and **D** would vary through formulas. New syntactical variables may be formed from old ones by adding primes or subscripts, and these new syntactical variables vary through the same expressions as the old ones. Two different syntactical variables occur in the same context do not necessarily represent different expressions, and they are **not** symbols of the language.

***Example* 1.1.** *Suppose that x is a symbol of the formal system F and suppose that it turns out that whenever we add the symbol x to the right of a formula of F, we obtain a new formula of F. If* **u** *has been agreed to be used as a syntactical variable, the fact can be expressed as follows: if* **u** *is a formula, then the expression obtained by adding x to the right of* **u** *is a formula.*

Definition. If **u** and **v** are two syntactical variables, **uv** stands for the expression obtained by juxtaposing **u** and **v**, that is, by writing down **v** immediately after writing **u**. Then the last part of the statement of the previous example can be shorten to “**u**x is a formula".

2

First-Order Theories

2.1 Functions and Predicates

Definition. A subset of the set of n-tuples in A is called an *n-ary predicate*. If P represents such predicate, then $P(a_1, \cdots, a_n)$ means that the n-tuple $(a_1, \cdots, a_n)$ is in P.

Definition. A certain set of objects in a mathematical axiom system that contains the objects "manipulated"(e.g. $\mathbb{N}$) is called the *universe* of the axiom system, and its elements are called the *individuals* of the system. Functions from the universe to the universe are called *individual functions*; predicates in the universe are called *individual predicates*.

Among the symbols needed in formalizing the axiom system are names for certain individuals, individual functions, and individual predicates.

Definition. The binary predicate whose first and second elements are the same element of a set, called the *equality predicate* in this set, is denoted by $=$.

2.2 Truth Functions

Definition. A *truth function* is the function from the set of *truth values* (usually $\mathbf{T}$ and $\mathbf{F}$) to the set of truth values. For example, the symbol $\wedge$ can be defined by a binary truth function $H_\wedge(a, b)$ that can be described by the equations

$$H_\wedge(\mathbf{T}, \mathbf{T}) = \mathbf{T}$$
$$H_\wedge(\mathbf{T}, \mathbf{F}) = H_\wedge(\mathbf{F}, \mathbf{T}) = H_\wedge(\mathbf{F}, \mathbf{F}) = \mathbf{F}$$

2.3 Variables and Quantifiers

Definition. A *free variable* is a notation that specifies a place in an expression where substitution may take place. A *bound variable* is a variable that no substitution (with an individual in the universe) can take place. For example, in the expression

$$\sum_{x=1}^{10} f(x, y)$$

we can change x to any variable other than y without altering the value of this expression. However, change x to 2 result in a meaningless expression. On the other hand, y can be substitute to any individual, such as 5, while the new expression still makes sense. Thus the value of this expression depends on y, the free variable, not the bound variable x.

Formally, an occurrence of $\mathbf{x}$ in $\mathbf{A}$ is *bound* in $\mathbf{A}$ if it occurs in a part of $\mathbf{A}$ of the form $\exists\mathbf{xB}$; otherwise, it is *free* in $\mathbf{A}$.

Definition. The process of *quantification* specifies the quantity of specimens in the universe that satisfies a open formula. A language element that generates a quantification is called a *quantifier*, such as $\forall$ and $\exists$.

2.4 First-Order Languages

Definition. A *first-order language* has as symbols the following:

1. The variables $x, y, z, w, x', y', \cdots$;
2. for each n, the n-ary function symbols and the n-ary predicate symbols;
3. the symbols $\neg$, $\vee$, and $\exists$.

Definition. A 0-ary function symbol is called a *constant*. A function symbol or a predicate symbol other than $=$ is called a *nonlogical* symbol; other symbols are called *logical* symbols.

Definition. With the symbols in a language, the *terms* can be defined by the following rules:

1. a variable is a term;
2. the combination of a n-ary and n terms is a term.

This combination is also called an *atomic formula*.

Definition. The *formulas* are defined as:

1. an atomic formula is a formula;
2. if $\mathbf{u}$ is a formula, $\neg\mathbf{u}$ is a formula;
3. if $\mathbf{u}$ and $\mathbf{v}$ are formulas, then $\vee\mathbf{uv}$ is a formula;
4. if $\mathbf{u}$ is a formula, then $\exists\mathbf{xu}$ is a formula.

The *height* of a formula is defined to be the number of occurrences of $\neg$, $\vee$, and $\exists$ in the formula.

Definition. A *designator* is an expression which is either a term or a formula. Every designator has the form $\mathbf{uv_1 \cdots v_n}$, where $\mathbf{u}$ is a symbol, the rest of $\mathbf{v}_i$ are designators, and n is determined by $\mathbf{u}$. n is called the *index* of $\mathbf{u}$. For example, if $\mathbf{u}$ is a variable, then $n = 0$; if $\mathbf{u}$ is a k-ary function symbol, then $n = k$; if $\mathbf{u}$ is $\exists$, then $n = 2$.

Two expressions are *compatible* if one of them can be obtained by adding some expression (possibly the empty expression) to the right end of the other.

Lemma. *If* $\mathbf{u}_1, \cdots, \mathbf{u}_n$, $\mathbf{u}'_1, \cdots, \mathbf{u}'_n$ *are designators and* $\mathbf{u}_1 \cdots \mathbf{u}_n$ *and* $\mathbf{u}'_1 \cdots \mathbf{u}'_n$ *are compatible, then* $\mathbf{u}_i$ *is* $\mathbf{u}'_i$ *for* $i = 1, \cdots, n$.

Proof. Prove by induction. $\square$

Theorem 2.4.1 (Formation Theorem)**.** *Every designator can be written in the form* $\mathbf{uv_1 \cdots v_n}$*, where* $\mathbf{u}$ *is a symbol, the rest of* $\mathbf{v}_i$ *are designators, and* n *is the index of* $\mathbf{u}$*, in one and only one way.*

Proof. $\mathbf{u}$ is the first symbol of it, so it must be uniquely determined, and thus n is unique. The previous lemma shows that if the combination of $\mathbf{u}$ and two sets of $\mathbf{v}_i$ are identical, the two sets of $\mathbf{v}_i$ must be identical. $\square$

Lemma. *Every occurrence of a symbol in a designator* $\mathbf{u}$ *begins with an occurrence of a designator in* $\mathbf{u}$.

Proof. Use induction on the length of $\mathbf{u}$. $\square$

Theorem 2.4.2 (Occurrence Theorem)**.** *Let* $\mathbf{u}$ *be a symbol of index* n*, and let* $\mathbf{v}_1, \cdots, \mathbf{v}_n$ *be designators. Then any occurrence of a designator* $\mathbf{v}$ *in* $\mathbf{uv_1 \cdots v_n}$ *is either all of* $\mathbf{uv_1 \cdots v_n}$ *or a part of one of the* $\mathbf{v}_i$.

Proof. If the initial symbol occurred in $\mathbf{v}$ is $\mathbf{u}$, then clearly $\mathbf{v}$ is compatible with $\mathbf{u}$, thus $\mathbf{v}$ is all of $\mathbf{uv_1 \cdots v_n}$.

If the initial symbol of occurrence of $\mathbf{v}$ is within $\mathbf{v}_i$. This symbol begins an occurrence of a designator $\mathbf{v}'$ in $\mathbf{v}_i$ by the previous lemma. Apparently $\mathbf{v}$ and $\mathbf{v}'$ are compatible, so $\mathbf{v}$ is $\mathbf{v}'$. Hence $\mathbf{v}$ is a part of $\mathbf{v}_i$. $\square$

Definition. The notation $\mathbf{b_x}[\mathbf{a}]$, which is a term, designates the expression obtained from $\mathbf{b}$ by replacing each occurrence of $\mathbf{x}$ by $\mathbf{a}$. The notation $\mathbf{A_x}[\mathbf{a}]$, which is a formula, designates the expression obtained from $\mathbf{A}$ by replacing each *free* occurrence of $\mathbf{x}$ by $\mathbf{a}$. We say that $\mathbf{a}$ is *substitutible for* $\mathbf{x}$ *in* $\mathbf{A}$ if for each variable $\mathbf{y}$ occurring in $\mathbf{a}$ no part of $\mathbf{A}$ of the form $\exists\mathbf{yB}$ contains an occurrence of $\mathbf{x}$ which is free in $\mathbf{A}$. The notations above can be easily extended to several variables, *e.g.* $\mathbf{b_{x_1,\cdots,x_n}}[\mathbf{a_1},\cdots,\mathbf{a_n}]$.

Here $\mathbf{u}$ is a binary predicate or function symbol.

Symbols	Meaning
$(\mathbf{A} \vee \mathbf{B})$	$\vee\mathbf{AB}$
$(\mathbf{A} \Rightarrow \mathbf{B})$	$(\neg\mathbf{A} \vee \mathbf{B})$
$(\mathbf{A} \wedge \mathbf{B})$	$\neg(A \Rightarrow \neg B)$
$(\mathbf{A} \Leftrightarrow \mathbf{B})$	$((\mathbf{A} \Rightarrow \mathbf{B}) \wedge (\mathbf{B} \Rightarrow \mathbf{A}))$
$\forall\mathbf{xA}$	$\neg\exists\mathbf{x}\neg\mathbf{A}$
$(\mathbf{aub})$	$(\mathbf{uab})$
$(\mathbf{a}\not{\mathbf{u}}\mathbf{b})$	$\neg(\mathbf{aub})$

Definition. $\neg\mathbf{A}$ is called the *negation* of $\mathbf{A}$; $\mathbf{A} \vee \mathbf{B}$ is the *disjunction* of $\mathbf{A}$ and $\mathbf{B}$, $\mathbf{A} \wedge \mathbf{B}$ the *conjunction*, $\mathbf{A} \Rightarrow \mathbf{B}$ the *implication*, $\mathbf{A} \Leftrightarrow \mathbf{B}$ the *equivalence*, $\exists\mathbf{xA}$ the *instantiation* of $\mathbf{A}$ by $\mathbf{x}$, and $\forall\mathbf{xA}$ the *generalization* of $\mathbf{A}$ by $\mathbf{x}$. $\forall$ is called the *universal quantifier*, and $\exists$ is called the *existential quantifier*.

2.5 Structures

Definition. Let L be a first-order language. A *structure* $\mathcal{S}$ for L consists of the following things:

1. A nonempty set $|\mathcal{S}|$, called the *universe* of $\mathcal{S}$, whose elements are called the *individuals* of $\mathcal{S}$.

2. For each n-ary function symbol $\mathbf{f}$ of L, and n-ary function $\mathbf{f}_\mathcal{S}$ from $|\mathcal{S}|$ to $|\mathcal{S}|$. *e.g.* for each constant $\mathbf{e}$ of L, $\mathbf{e}_\mathcal{S}$ is an individual of $\mathcal{S}$.

3. For each n-ary predicate symbol $\mathbf{p}$ of L other then $=$, and n-ary predicate $\mathbf{p}_\mathcal{S}$ in $|\mathcal{S}|$.

Definition. Let $\mathcal{S}$ be a structure for L. For each individual of it we choose a new constant called the *name* of this individual. Different names are chosen for different individuals. The first-order Language obtained from L by adding all the names of individuals of $\mathcal{S}$ is designated by $L(\mathcal{S})$.

Definition. An expression is *variable-free* if it contains no variables. The individual $\mathcal{S}(\mathbf{a})$ of $\mathcal{S}$ for each variable-free term $\mathbf{a}$ of $L(\mathcal{S})$ is defined as follows: if $\mathbf{a}$ is a name, $\mathcal{S}(\mathbf{a})$ is the individual of which $\mathbf{a}$ is the name; if $\mathbf{a}$ is not a name, then since it is variable-free it must be $\mathbf{fa_1} \cdots \mathbf{a_n}$ with $\mathbf{f}$ as a function symbol of L, and we then let $\mathcal{S}(\mathbf{a})$ be $\mathbf{f}_{\mathcal{S}}(\mathcal{S}(\mathbf{a_1}), \cdots, \mathcal{S}(\mathbf{a_n}))$.

Definition. A formula $\mathbf{A}$ is *closed* if no variable is free in $\mathbf{A}$. Intuitively, this formula has no "open" interpretation so that in any circumstances the meaning of it is fixed.

We now define a truth value $\mathcal{S}(\mathbf{A})$ for each closed formula $\mathbf{A}$ in $L(\mathcal{S})$.

$\mathbf{A} = ...$	Truth Value
$\mathbf{a} = \mathbf{b}$	$\mathcal{S}(\mathbf{A}) = T \Leftrightarrow \mathcal{S}(\mathbf{a}) = \mathcal{S}(\mathbf{b})$
$\mathbf{pa_1} \cdots \mathbf{a_n}$, where $\mathbf{p}$ is not $=$	$\mathcal{S}(\mathbf{A}) = T \Leftrightarrow \mathbf{p}_{\mathcal{S}}(\mathcal{S}(\mathbf{a_1}), \cdots, \mathcal{S}(\mathbf{a_n}))$
$\neg\mathbf{B}$	$H_\neg(\mathcal{S}(\mathbf{B}))$
$\mathbf{B} \vee \mathbf{C}$	$H_\vee(\mathcal{S}(\mathbf{B}), \mathcal{S}(\mathbf{C}))$
$\exists\mathbf{xB}$	$\mathbf{S}(\mathbf{A}) = T$ iff $\mathcal{S}(\mathbf{B_x}[\mathbf{i}]) = T$ for some $\mathbf{i}$ in $L(\mathcal{S})$

Now we have

1. $\mathcal{S}(\mathbf{A_1} \vee \cdots \vee \mathbf{A_n}) = T$ iff $\mathcal{S}(\mathbf{A}_i) = T$ for at least one i.
2. $\mathcal{S}(\mathbf{A_1} \wedge \cdots \wedge \mathbf{A_n}) = T$ iff $\mathcal{S}(\mathbf{A}_i) = T$ for all i.
3. $\mathcal{S}(\forall\mathbf{xA}) = T$ iff $\mathcal{S}(\mathbf{A_x}[\mathbf{i}]) = T$ for every $\mathbf{i}$ in $L(\mathcal{S})$.

Definition. If $\mathbf{A}$ is a formula of L, and $\mathcal{S}$-*instance* of $\mathbf{A}$ is a closed formula of the form $\mathbf{A}[\mathbf{i_1}, \cdots, \mathbf{i_n}]$ in $L(\mathcal{S})$. A formula $\mathbf{A}$ of L is *valid* in $\mathcal{S}$ if $\mathcal{S}(\mathbf{A}') = T$ for every $\mathcal{S}$-instance $\mathbf{A}'$ of $\mathbf{A}$.

Lemma. *Let $\mathcal{S}$ be a structure for L; $\mathbf{a}$ a variable-free term in $L(\mathcal{S})$; $\mathbf{i}$ the name of $\mathcal{S}(\mathbf{a})$. If $\mathbf{b}$ is a term of $L(\mathcal{S})$ in which no variable except $\mathbf{x}$ occurs, then $\mathcal{S}(\mathbf{b_x}[\mathbf{a}]) = \mathcal{S}(\mathbf{b_x}[\mathbf{i}])$. If $\mathbf{A}$ is a formula of $L(\mathcal{S})$ in which no variable except $\mathbf{x}$ is free, then $\mathcal{S}(\mathbf{A_x}[\mathbf{a}]) = \mathcal{S}(\mathbf{A_x}[\mathbf{i}])$*

Proof. If $\mathbf{b}$ is a name or a variable (in this case it would only involve $\mathbf{x}$), the first conclusion holds obviously. If $\mathbf{b}$ is a function symbol $\mathbf{fb_1} \cdots \mathbf{b_n}$, we would have

$$\begin{aligned}
\mathcal{S}(\mathbf{b_x}[\mathbf{a}]) &= \mathcal{S}(\mathbf{fb_1}[\mathbf{a}] \cdots \mathbf{b_n}[\mathbf{a}]) \\
&= \mathbf{f}(\mathcal{S}(\mathbf{b_1}[\mathbf{a}]) \cdots \mathcal{S}(\mathbf{b_n}[\mathbf{a}])) \\
&= \mathbf{f}(\mathcal{S}(\mathbf{b_1}[\mathbf{i}]) \cdots \mathcal{S}(\mathbf{b_n}[\mathbf{i}])) \quad \text{by the induction hypothesis} \\
&= \mathcal{S}(\mathbf{fb_1}[\mathbf{i}] \cdots \mathbf{b_n}[\mathbf{i}]) \\
&= \mathcal{S}(\mathbf{b_x}[\mathbf{i}])
\end{aligned}$$

For a formula $\mathbf{A}$, the assertion can be verified for the case when $\mathbf{A}$ is: $\mathbf{b} = \mathbf{c}$, $\neg\mathbf{B}$, $\mathbf{B} \vee \mathbf{C}$, and $\exists\mathbf{yB}$(we may suppose $\mathbf{y}$ is not $\mathbf{x}$ or otherwise $\mathbf{A_x}[\mathbf{a}]$ and $\mathbf{A_x}[\mathbf{i}]$ would both be $\mathbf{A}$), in a similar way. □

2.6 Logical Axioms and Rules

Definition. Let L be a first-order language. A *propositional axiom* is a formula of the form $\neg\mathbf{A} \vee \mathbf{A}$. A *substitution axiom* is a formula of the form $\mathbf{A_x}[\mathbf{a}] \Rightarrow \exists\mathbf{x}\mathbf{A}$. An *identity axiom* is a formula of the form $\mathbf{x} = \mathbf{x}$. An *equality axiom* is a formula of the form

$$\mathbf{x_1} = \mathbf{y_1} \Rightarrow \cdots \Rightarrow \mathbf{x_n} = \mathbf{y_n} \Rightarrow \mathbf{f}\mathbf{x_1}\cdots\mathbf{x_n} = \mathbf{f}\mathbf{y_1}\cdots\mathbf{y_n}$$

or of the form

$$\mathbf{x_1} = \mathbf{y_1} \Rightarrow \cdots \Rightarrow \mathbf{x_n} = \mathbf{y_n} \Rightarrow \mathbf{p}\mathbf{x_1}\cdots\mathbf{x_n} = \mathbf{p}\mathbf{y_1}\cdots\mathbf{y_n}$$

A *logical axiom* is a formula which is a propositional axiom, a substitution axiom, an identity axiom, or an equality axiom.

The logical axioms are valid. The proofs for a propositional axiom, a identity axiom, and a equality axiom is obvious. An $\mathcal{S}$-instance of a substitution axiom has the form $\mathbf{A_x}[\mathbf{a}] \Rightarrow \exists\mathbf{x}\mathbf{A}$. Suppose that $\mathcal{S}(\mathbf{A_x}[\mathbf{a}] \Rightarrow \exists\mathbf{x}\mathbf{A}) = F$. Then $\mathcal{S}(\mathbf{A_x}[\mathbf{a}]) = T$ and $\mathcal{S}(\exists\mathbf{x}\mathbf{A}) = F$. If $\mathbf{i}$ is the name of $\mathcal{S}(\mathbf{a})$, the latter implies that $\mathcal{S}(\mathbf{A_x}[\mathbf{i}]) = F$ while the former with the previous lemma implies that $\mathcal{S}(\mathbf{A_x}[\mathbf{i}]) = T$, a contradiction.

We now introduce five rules of inference (note that these rules are finite).

1. *Expansion Rule*: Infer $\mathbf{B} \vee \mathbf{A}$ from $\mathbf{A}$.
2. *Contraction Rule*: Infer $\mathbf{A}$ from $\mathbf{A} \vee \mathbf{A}$.
3. *Associative Rule*: Infer $(\mathbf{A} \vee \mathbf{B}) \vee \mathbf{C}$ from $\mathbf{A} \vee (\mathbf{B} \vee \mathbf{C})$.
4. *Cut Rule*: Infer $\mathbf{B} \vee \mathbf{C}$ from $\mathbf{A} \vee \mathbf{B}$ and $\neg\mathbf{A} \vee \mathbf{C}$.
5. $\exists$-*Introduction Rule*: If $\mathbf{x}$ is not free in $\mathbf{B}$, infer $\exists\mathbf{x}\mathbf{A} \Rightarrow \mathbf{B}$ from $\mathbf{A} \Rightarrow \mathbf{B}$.

We now prove that all the rules are valid.

Proof. content... $\square$