A Quick Introduction to Mathematical Logic

Li Jiatu[*]

May 12, 2020

Contents

[*]lijt19@mails.tsinghua.edu.cn

1 Introduction

The study of logic dates back to ancient Greek philosophers Sokrates and Aristotle. A famous example of logic that is used in almost all textbooks is the following one.

All humans are mortal; Sokrates is a human; therefore Sokrates is mortal.

In a word, a logic system is a language that can describe propositions.

In order to study the property of a logic system, we need to formally define the syntax to form a proposition (more preciously, a formula) and semantic meaning of it. We will do such work in Section 2 and 3. In Section 2, we will describe the propositional logic, which can express the relationship between atomic propositions, i.e. φ imples ψ. Then we will extend the language with quantifiers to obtain first-order logic, so that we can express the statements like "all humans are mortal".

It is clear that some sentences is "always true", for example, φ imples φ itself. We will not be satisfied if our language can only *state* a proposition but can not *prove* it. The example of Sokrates gives an intuitive description of reasoning with logic system, and what we will do is to describe a formal proof system, in order to study the characteristic of reasoning. Furthermore, we are interested in the properties of proof systems, for example, the completeness, which means that each "true" formula in the language can be proved in the proof system. We will go though these things for propositional logic and first-order logic in Section 4 and 5.

In Sections 6, we will consider a more exciting concept about first-order logic, which is called model. A model of a set of sentencs Δ is a mathematical structure such that all the sentences in Δ are satisfied. We will show the deep relationship between the completness of logic system and the existence of model.

We will close our discussion about logic with a famous result in mathematical logic, which is called Gödel's Incompleteness Theorem. It shows that any expressive theory, as long as it contains the arithmetic theory called Peano Arithmetic, is not complete. The influence of the result is huge in both mathematics, logic and computer science (even though the first computer has not been built at that time).

2 Propositional Logic

2.1 Intuitive Description

Suppose we have a random variable X which takes its value uniformly from $\{1, 2, \dots, 6\}$. We do an experiment to obtain a concrete value x, but before observing it we may make some guess about the value. For example, we have some propositions $\varphi_2, \varphi_3, \dots, \varphi_6$, where φ_i stands for the statement that $x \geq i$.

Although we can not determine whether each of the statements is true or not before observing x, we are quite sure that at least one of the propositions is true if $x \neq 1$. We may say, in natural language, that "at least one of the following statements is true, $\varphi_2, \varphi_3, \dots, \varphi_6$, if $x \neq 1$". Similarly, if φ_3 is true, we know that φ_2 must be true, which may be expressed with "φ_3 implies φ_2". This example shows that, if we have a set of statements, the relationship between them is quite interesting. Furthermore, we are more interested in the relationships that are widely used in our lives, that is

1. φ *and* ψ, meaning that both of the two statements are true;
2. φ *or* ψ, meaning that at least one of the two statements is true;
3. φ *implies* ψ, meaning that if φ is true, we must have ψ is true.

Moreover, we need to express the statement which stands for the negative of another statement, that is

- *not* φ, meaning that φ is not true.

The statements formed by relations (or more formally, connectives) and *atomic* propositions form the language of propositional logic.

2.2 Syntax

Now we wound like to describe the logic system in a more mathematical way. Suppose L is a set of symbols which is called the vocabulary of the language, a string φ is called a formula, if

1. $\varphi = p$, where $p \in L$;
2. $\varphi = (\psi \land \rho)$, where ψ and ρ are formulas;
3. $\varphi = (\psi \lor \rho)$, where ψ and ρ are formulas;
4. $\varphi = (\psi \to \rho)$, where ψ and ρ are formulas;
5. $\varphi = (\neg\psi)$, where ψ is a formula.

We may ignore the parentheses if it is not confusing.

Note that this definition is inductive, meaning that we use the definition of formula in the definition of itself. This will not lead to a circular definition (so we will say that the definition is well-founded), since if $\varphi = \psi \land \rho$, the length of ψ and ρ is shorter than the length of φ, so that we can enumerate all strings in increasing order of length to determine whether each of the strings is a formula.

Since the formulas are inductively defined, we can also define some functions about formulas inductively. For example, we may define the length of a formula as

1. $|p| = 1$;
2. $|\varphi \wedge \psi| = |\varphi| + |\psi| + 1$;
3. $|\varphi \wedge \psi| = |\varphi| + |\psi| + 1$;
4. $|\varphi \vee \psi| = |\varphi| + |\psi| + 1$;
5. $|\varphi \to \psi| = |\varphi| + |\psi| + 1$;
6. $|\neg\varphi| = |\varphi| + 1$.

Also we can define the atomic propositions used in a formula, which is left as exercise.

One of another important thing about inductive definition is that, we can do inductive proof for the property about formulas. Suppose we have a property $P(\varphi)$ about formula φ, for instance, $P(\varphi)$ stands for the statement that φ contains only finite number of atomic propositions. We may prove the proposition the following way:

1. show that each formula formed by single atomic proposition satisfies P; and
2. suppose ψ and ρ satisfying P, prove that the all formulas formed by ψ, ρ and conjunction satisfy P.

For this concrete $P(\varphi)$, both of the steps is straightforward.

2.3 Semantics

Now we will formally define the semantics of the formulas. Go back to the example of random variable in Section 2.1. Before observing x, we may state lots of propositions about the relationship of $\varphi_2, \varphi_3, \ldots, \varphi_n$. Some of them are always true, for example, φ_2 imples φ_3; some of them are always false, for example, φ imples that φ_3 is not true. In general, we can not be determined whether a proposition is true until observing x. This tells us that the truth value of a formula is meaningful only if we know the truth value of atomic formulas.

We may restate this observation in a more formal way. We define the valuation or interpretation of atomic propositions as a function $V : L \to \{T, F\}$, where T stands for true and F stands for false. For simplicity, we may also use 0 and 1 instead of F and T respectively. As an extension, we may define the interpretation of a formula V^* inductively.

1. $V^*(p) = V(p)$ for all $p \in L$;
2. $V^*(\varphi \wedge \psi) = T$ if and only if both $V^*(\varphi)$ and $V^*(\psi)$ equals to T;
3. $V^*(\varphi \vee \psi) = T$ if and only if at least one of $V^*(\varphi)$ and $V^*(\psi)$ equals to T;
4. $V^*(\varphi \to \psi) = T$ if and only if $V^*(\varphi) = F$, or $V^*(\psi) = T$;
5. $V^*(\neg\varphi) = T$ if and only if $V^*(\varphi) = F$.

If it is clear in context, we will simply use V to represent V^*. Also, we will use $V \models \varphi$ to denote $V(\varphi) = T$.

Intuitively, the connectives $\wedge$, $\vee$, $\to$ and $\neg$ represents the relationship “and”, “or”, “imply” and “not” respectively. This definition is just a restatement of the intuitive meaning of these connectives, so that we can formally prove the property of the logic system.

Sometimes we may use the tool called truth table to express the meaning of a connective. For instance, the truth table of $\wedge$ is given as follow.

φ	ψ	$\varphi \wedge \psi$
T	T	T
T	F	F
F	T	F
F	F	F

Table 1: Truth table of $\wedge$.

This table should be read in rows. For example, the first row of the table means that if $V(\varphi) = T$ and $V(\psi) = F$, then we have $V(\varphi \wedge \psi) = T$. Since there are only $2^2 = 4$ possible configurations for φ and ψ, the table have four columns. It will be a good practice to write down the truth table for all connectives in propositional logic.

2.4 Truth Table of Formula

By the definition of valuation, we may notice that the formulas, in some sense, like the connectives with many arguments. Suppose φ is a formula containing atomic propositions $p_1, p_2, \ldots, p_n$, we know the truth value of φ if the truth value of $p_1, p_2, \ldots, p_n$ are fixed. We may draw a truth table with 2^n rows and $n+1$ columns, to describe the truth value of φ under 2^n possible configurations of $p_1, \ldots, p_n$.

We will develop several concepts about formulas with truth table of it. A formula is called a tautology (or it is logically true), if each row of the last column of its truth table is T, i.e. it is true no matter how we interprete $p_1, \ldots, p_n$. We will use $\vDash \varphi$ to denote φ is a tautology. A formula is called contradictory, if each row of the last column is F. A formula is called satisfiable, if at least one row of the last column of its truth table is T. Two formulas are called logically equivalent, denoted as $\varphi \equiv \psi$, if they have the same truth table.

It is clear, by the knowledge of combinatorics, that there are at most 2^{2^n} logically distinct formulas containing n atomic propositions. A natural question is, if it is possible to construct all 2^{2^n} logically distinct formulas with our conjuctions in propositional logic, and the answer the this question is yes.

2.5 Complete Set of Conjuctions

A set of connectives is called complete, if they can generate all logically distinct formulas.

Theorem 1. *$S = \{\neg, \wedge, \vee\}$ is a complete set of connectives.*

Proof. Suppose $p_1, p_2, \ldots, p_n$ are atomic propositions and $P(p_1, p_2, \ldots, p_n)$ is any truth table, i.e. $P(x_1, x_2, \ldots, x_n)$ represents the last column of the row where $p_i = x_i$. Suppose P is unsatisfiable, we will simply generate $\neg(p_1 \to p_1)$. Otherwise, if P contains k rows such that the last column of it is T, for each of them denoted as $r_i = (x_1, x_2, \ldots, x_n)$, we will generate a formula φ_i such that

$$\varphi_i = \psi_{i1} \wedge \psi_{i2} \wedge \cdots \wedge \psi_{in},$$

where ψ_{ij} is defined as

1. if $x_j = T$, $\psi_{ij} = p_j$;
2. otherwise, $\psi_{ij} = \neg p_j$.

Let φ be the formula

$$\varphi_1 \vee \varphi_2 \vee \cdots \vee \varphi_k.$$

Now we will show that $P(x_1, x_2, \ldots, x_n)$ is T if and only if $V(\varphi) = T$, under any interpretation $V(p_i) = x_i$.

Suppose $V(\varphi) = T$ under the interpretation $V(p_i) = x_i$, by the definition of $\vee$, we know that at least one of φ_i satisfies $V(\varphi_i) = T$. Then, by the definition of φ and $\wedge$, we know that each one of $\psi_{i1}, \psi_{i2}, \ldots, \psi_{in}$ is true under the interpretation. This means that, the valuation of $p_1, p_2, \ldots, p_n$, denoted as $x_1, x_2, \ldots, x_n$ is a row with T in its last column, so that $P(x_1, \ldots, x_n)$ is T.

The other side of the proof is left as exercise. $\square$

Clearly, the formula we have constructed in the proof is very special, which is called a disjunctive normal form. Similarly, we can construct a conjunctive normal form by considering the rows of truth table with F in its last column. For simplicity, the disjunctive normal form of a contradictory formula and the conjunction normal form of a tautology is represented as $\vee_0$ and $\wedge_0$ in special. Then the theorem above can be restated as follow.

Theorem 2. *Each formula in propositional logic has an equivalent disjunctive or conjunctive normal form.*

We may also find some other complete set of connectives, for example, $\{\neg, \vee\}$ is also complete. This is because we can simulate $\varphi \wedge \psi$ as $\neg(\neg\varphi \vee \neg\psi)$. Similarly, $\{\neg, \rightarrow\}$ is complete, since we can simulate $\varphi \vee \psi$ as $\neg(\neg\varphi \rightarrow \psi)$. The most amazing set of complete connectives has only one binary conjuction $\downarrow$, whose truth table is given as follow.

φ	ψ	$\varphi \downarrow \psi$
T	T	F
T	F	T
F	T	T
F	F	T

Table 2: Truth Table of $\downarrow$.

It may take some time to show that it is complete itself.

2.6 Equivalence

Now we wound like to develop some useful equations about logical equivalence to provide an intuition about logic. Before doing this, it will be good to define the rules to reduce the use of parentheses. Firstly, $\neg$ has highest priority, meaning that $\neg\varphi \wedge \psi$ will be considered as $(\neg\varphi) \wedge \psi$. Similarly, $\wedge$ has higher priority than $\vee$, and $\vee$ has higher priority than $\rightarrow$.

$\wedge$ and $\vee$ is left associated, meaning that $\varphi \wedge \psi \wedge \rho$ is consider to be $(\varphi \wedge \psi) \wedge \rho$. $\to$ is right associated, meaning that $\varphi \to \psi \to \rho$ is considered to be $\varphi \to (\psi \to \rho)$. This is quite similar to the case in the ordinary arithmetic.

The first equation we will show, which we have already known, is that

$$\neg(\varphi \wedge \psi) \equiv \neg\varphi \vee \neg\psi, \tag{1}$$
$$\neg(\varphi \vee \psi) \equiv \neg\varphi \wedge \neg\psi. \tag{2}$$

This equation is called de Morgen's law. It provides a way to define $\wedge$ and $\vee$ from each other.

The second equation tells us how to define $\to$ with $\neg$ and $\vee$.

$$\varphi \to \psi \equiv \neg\varphi \vee \psi. \tag{3}$$

If you still remember the definition of $\to$, you may immediately find that this is just a restatement of its definition in propositional logic.

The third one, which is called double negative elimination, is one of the most important property of $\neg$.

$$\neg\neg\varphi \equiv \varphi. \tag{4}$$

Such equivalence is also widely used when we do informal logical reasoning. Note that this rule is closely related to the fact called excluded middle, which is

$$\vDash \varphi \vee \neg\varphi. \tag{5}$$

This is to say, a proposition is either true or false, which is quite straightforward.

The last one is the distributive law about $\wedge$ and $\vee$.

$$\varphi \vee (\psi \wedge \rho) \equiv (\varphi \vee \psi) \wedge (\varphi \vee \rho), \tag{6}$$
$$\varphi \wedge (\psi \vee \rho) \equiv (\varphi \wedge \psi) \vee (\varphi \wedge \rho). \tag{7}$$

This rule is closely related to the algebraic property of $\wedge$ and $\vee$. We refer the readers with great interests to learn the lattice theory and Boolean algebra.

It is recommended, especially for the readers who are not familiar to logic, to do some exercise about logical equivalence, in order to build up the intuition about propositional logic.

2.7 Logical Consequence

The last thing we will talk about in this section is the logical consequence, which is like a "one way" equivalence. Suppose Γ is a set of formulas and φ is a formula, we define φ is a logical consequence of Γ, if there is no valuation in which

1. each sentences in Γ is true, and
2. φ is false.

We will denote this with the notation $\Gamma \vDash \varphi$, which is consistent with the notation of tautology. An useful criteria of logical consequence, which may be useful in future, is given by the following lemma.

Lemma 1. $\Gamma \vDash \varphi$ *if and only if* $\Gamma \cup \{\neg\varphi\}$ *is unsatisfiable, i.e. there is no valuation in which each sentence in it is true.*

The proof of the lemma is straightforward and is left as exercise.

We may show some useful properties of logical consequence. The first one is called the hypothetical syllogism. Let us define $\Gamma_1 \vDash \Gamma_2$ as $\Gamma_1 \vDash \varphi$ for all $\varphi \in \Gamma_2$.

Lemma 2. *Suppose* Γ_1, Γ_2 *and* Γ_3 *are sets of formulas. If* $\Gamma_1 \vDash \Gamma_2$*,* $\Gamma_2 \vDash \Gamma_3$*, then* $\Gamma_1 \vDash \Gamma_3$*. In particular, if* $\varphi \vDash \psi$*,* $\psi \vDash \rho$*, we know that* $\varphi \vDash \rho$*.*

Proof. For all valuation in which each sentence in Γ_1 is true, since $\Gamma_1 \vDash \Gamma_2$, we know that each sentence in Γ_2 is true. Then since $\Gamma_2 \vDash \Gamma_3$, we know that each sentence in Γ_3 is true. Then by definition, we know that $\Gamma_1 \vDash \Gamma_3$. In particular, let $\Gamma_1 = \{\varphi\}$, $\Gamma_2 = \{\psi\}$ and $\Gamma_3 = \{\rho\}$, we get the case for single formula. □

One the the most important properties of logical consequence is its relationship with $\to$, the logical implication. This is shown in the following theorem.

Theorem 3. *Suppose* Γ *is a set of formulas and* φ, ψ *are formulas, then* $\Gamma, \varphi \vDash \psi$ *if and only if* $\Gamma \vDash \varphi \to \psi$*.*

Proof. Suppose $\Gamma, \varphi \vDash \psi$, by the criteria of logical consequence, we know that $\Gamma, \varphi, \neg\psi$ is unsatisfiable, which means exactly that $\Gamma, \neg(\varphi \to \psi)$ is unsatisfiable, where $\neg(\varphi \to \psi) \equiv \varphi \wedge \neg\psi$. Again with the criteria, we know that $\Gamma \vDash \varphi \to \psi$.

Conversely, suppose $\Gamma \vDash \varphi \to \psi$, we know that $\Gamma, \varphi \to \psi$ is unsatisfiable, so that $\Gamma, \varphi, \neg\psi$ is unsatisfiable. By the criteria, we know that $\Gamma, \varphi \vDash \psi$. □

Note that we define $\Gamma, \varphi \vDash \psi$ as $\Gamma \cup \{\varphi\} \vDash \psi$. During the proof, we use the fact that $\Gamma \cup \{\varphi \wedge \psi\}$ is satisfiable if and only if $\Gamma \cup \{\varphi, \psi\}$ is satisfiable, which is quite straightforward.

3 First-Order Logic

3.1 Intuitive Description

It is clear that the propositional logic is *not* sufficiently expressive. It can express non of the following statements.

1. *All humans are mortal.*
2. *Some peaple have died, but he is still alive*[1].
3. *For all* $\epsilon > 0$*, there exists* $\delta > 0$*, for all* x *such that* $|x - x_0| < \delta$*, we have* $|f(x) - f(x_0)| < \epsilon$*.*

It's not hard to see that these statements are restricted by quantifiers. The quantifiers identify the domain that the statement is true, for instance, ***some*** *people have died* means that among the set of people, there exists someone who have dead; ***all*** *humans are mortal* means that everyone among the set of people is mortal. In general, we are especially interested in two kinds of quantifiers, the universal quantifier and the existential quantifier.

[1]Some People, by Zang Kejia, 1949.

Informally, we may already familiar to do reasoning with quantifiers. When we say *all humans are mortal*, we immediately know that Sokrates, a human, must be mortal; when we say *some peaple have died, but he is still alive*, we can conclude that it is not the case that everyone dead is not alive[2].

Just like we have done for propositional logic, we use some symbols, $\forall$ and $\exists$, to represent universal and existential quantifiers, respectively. Moreover, we will introduce the <u>predicates</u>, which describes the properties of an "element". Suppose $\varphi(x)$ denotes that "x is a human", $\psi(x)$ denotes that "x is mortal", then the sentence *all humans are mortal* can be expressed as $(\forall x)(\varphi(x) \to \psi(x))$.

It will be good to identify the difference of x and φ in the sentence $(\forall x)\varphi(x)$. x is called a <u>variable</u> and $\varphi(*)$ is called a <u>predicate</u>. They are completely different things in <u>first-order logic</u>, i.e. we can not state a sentence like $(\forall \varphi)(\forall x)\varphi(x)$. Besides variables and predicates, the building blocks of first-order logic contains <u>constants</u>, <u>functions</u> and a special binary quantifier called the <u>identity</u>. We will formally describe the syntax of first-order logic in the following section.

3.2 Syntax

The vocabulary of first-order logic contains an infinite supply of

1. constants, denoted by $a, b, c, \ldots$;
2. (n ary) predicates, denoted by $P, Q, \ldots$;
3. (n ary) functions, denoted by $f, g, h, \ldots$.

For a given vocabulary L, we can define the <u>term</u> inductively, as

1. a variable is a term;
2. a constant is a term;
3. suppose f is a function with n arguments, and $t_1, t_2, \ldots, t_n$ are terms, then $f(t_1, t_2, \ldots, t_n)$ is a term.

Based on terms, we can define an <u>atomic formula</u> as

1. suppose P is a predicate with n arguments and $t_1, \ldots, t_n$ are terms, $P(t_1, \ldots, t_n)$ is an atomic formula;
2. suppose t_1 and t_2 are terms, $t_1 = t_2$ is an atomic formula, where $=$ is the identity predicate.

Moreover, we combine the atomic formulas as a formula, defined as

1. an atomic formula is a formula;
2. suppose φ, ψ are formulas, $\varphi \wedge \psi$, $\varphi \vee \psi$, $\neg\varphi$, $\varphi \to \psi$ are all formulas (where parentheses should be added if it is ambiguity);
3. suppose φ is a formula, $(\forall x)\varphi$ and $(\exists x)\varphi$ are formulas.

Clearly, the syntax of the first-order logic is much more complex than the that of the propositional logic. The main reason, as we have mentioned, is the difference between variables (and constants) with predicates. New language allows us to describe the property of an "object" of claim a statement over the set of "objects", which is not possible in propositional logic.

[2]Of course, this statement is true only when we define "alive" in some unusual way.

In order to save us from parentheses, we define that the priority of $\forall$ and $\exists$ are higher than any conjuctions, and the priority between conjuctions inherit from proposition logic. In particular, if there is no ambiguity, we will omit the parentheses around the quantifiers, i.e. we will simply use $\forall x P(x) \wedge Q$ to denote $((\forall x)P(x)) \wedge Q(x)$.

3.3 Semantics

Now we will develop the semantic of first-order logic. Recall that in the propositional logic, in order to determine the truth value of a formula, we need to fix an interpretation of atomic predicates. In first-order logic, such interpretation is called a model.

We need some basic definitions to deliver the idea of model. In a formula of the first-order logic, there may be some variables that is not controlled by any quantifier, i.e. y in the formula $\forall x P(x, y)$. Such variables is called free variables of the formula, and can be easily defined inductively. It is clear that a formula is a statement only if there is no free variable, thus the formulas with no free variable are called sentences.

We may denote a formula with n free variables $x_1, \ldots, x_n$ as $\varphi(x_1, \ldots, x_n)$. The substitution of x_i with ψ_i, denoted as

$$\varphi(x_1 \mapsto \psi_1, x_2 \mapsto \psi_2, \ldots, x_n \mapsto \psi_n),$$

is the formula obtained by replacing each free occurrence of x_i with ψ_i for $1 \leq i \leq n$. If ψ_i does not contain free variables, the formula becomes a sentence. The substitution can be defined inductively in which we need to carefully deal with the duplication of variable name. For instance in the formula below, the first x, marked with $\star$ is bounded by the universal quantifier; but the second x, marked with $\circ$ is a free variable.

$$\forall x A(x^\star) \to x^\circ.$$

In a substitution of x with ψ, only free occurrence of x need to be replaced by ψ, thus we get

$$\forall x A(x) \to \psi.$$

A model for vocabulary L, denoted as $\mathcal{M}$, is a pair (M, I). M is a non-empty set which is called the domain or universe, and I is a function that maps each symbol (i.e. constants, predicates and functions) from L into M. For simplicity, we may use $c^\mathcal{M}$, $P^\mathcal{M}$, $f^\mathcal{M}$ to denote $I(c)$, $I(P)$ and $I(f)$ respectively.

In the domain M, there are three kinds of objects:

1. elements, which is the interpretation of constants $c_1^\mathcal{M}, c_2^\mathcal{M}, \ldots$;
2. (n ary) relations among elements in the domain, which is the interpretation of predicates $P_1^\mathcal{M}, P_2^\mathcal{M}, \ldots$;
3. (n ary) functions among elements in the domain, which is the interpretation of functions $f_1^\mathcal{M}, f_2^\mathcal{M}, \ldots$.

Just like the definition of syntax, we will firstly define the value of a term without variable. For a term t without variable,

1. if t is some constant c, the value of t, denoted as $t^\mathcal{M}$ is defined as $c^\mathcal{M}$;

2. if t is some function $f(t_1, \ldots, t_k)$, the value of t, denoted as $t^{\mathcal{M}}$ is defined as $f^{\mathcal{M}}(t_1^{\mathcal{M}}, t_2^{\mathcal{M}}, \ldots, t_n^{\mathcal{M}})$.

Again, this definition is inductive over the structure of t.

Then we define the truth value of an atomic formula φ without variable as,

1. if φ is some predicate $P(t_1, \ldots, t_n)$, where $t_1, \ldots, t_n$ contain no variable, then the formula is true if and only if the relation $P^{\mathcal{M}}$ holds for elements $t_1^{\mathcal{M}}, t_2^{\mathcal{M}}, \ldots, t_n^{\mathcal{M}}$;
2. if φ is the identity $t_1 = t_2$, where t_1 and t_2 contain no variable, then the formula is true if and only if $t_1^{\mathcal{M}}$ and $t_2^{\mathcal{M}}$ are the same element in the domain.

Finally we define the truth value of a sentence φ (i.e. there is no free variable) as

1. if φ is $\neg\psi$, $\psi \wedge \rho$, $\psi \vee \rho$ or $\psi \to \rho$, where ψ and ρ has no free variable, then the truth definition is the same as that in propositional logic. For example, $\psi \wedge \rho$ is true if and only if both ψ and ρ are true;
2. if φ is $\forall x\psi(x)$, where x is the only free variable of ψ, then φ is true if and only if for all $c^{\mathcal{M}}$ in the domain, the sentence $\psi(x \mapsto c)$ is true;
3. if ψ is $\exists x\psi(x)$, where x is the only free variable of ψ, then φ is true if and only if there exists some $c^{\mathcal{M}}$ in the domain, such that the sentence $\psi(x \mapsto c)$ is true.

Some readers may be confused by the long definition of semantics, since it looks like a formal version of "circular definition". When we define the semantics of $\forall x\psi(x)$, we can not say a word about what is the meaning of "for all $c^{\mathcal{M}}$" used in our definition, so that it seems that we are defining the semantics of sentences with its own meaning. In some sense, this is correct. When we define the meaning of first-order logic, it is not avoidable to use another language, for example, English, to state our definition. This language is sometimes called the meta-language. If we want to define the semantics of our meta-language, we must have some meta-meta-language, and the process will eventually end at somewhere. Mathematics can not be developed from zero, there must be something (may be from human intelligence, if you are satisfied) that remains unexplained.

The true value of such definition is that, we can study the property of the formally defined language. It is like a programming language in mathematics world: although most of the computer scientists (mathematicians) work with pseudocode (informal language), knowing that there exists a programming language (mathematical logic) help them do reasoning more confidently.

3.4 Important Concepts

We will now go through the definition of important concepts for first-order logic. A sentence φ is called satisfiable, if there exists a model in which φ is true. Moreover, a set of sentences Γ, or more fancy, a theory Γ is called satisfiable, if there exists a model in which each sentence in Γ is true.

Similarly, a sentence is logically true, if it is true in all models; two sentence φ and ψ are called equivalent, if their truth value are the same in all models. Such definition can also be extended to the theory (i.e. the set of sentences). A sentence

φ is called the logical consequence of Γ, if $\Gamma \cup \{\neg\varphi\}$ is not satisfiable. All of the notations are the same at the propositional logic.

Just like in propositional logic, we can present the following theorem about equivalence, satisfiability and logical consequence.

Theorem 4. $\Gamma_1 \vDash \Gamma_2$ *if and only if* $\Gamma_1 \to \Gamma_2$ *is logically true;* $\Gamma_1 \equiv \Gamma_2$ *if and only if* $\Gamma_1 \leftrightarrow \Gamma_2$ *is logically true.*

Note that $\varphi \leftrightarrow \psi$ is defined as $(\varphi \to \psi) \wedge (\psi \to \varphi)$. The proof of the theorem is left as exercise. We will also show a more "first-order style" theorem which is very similar to this theorem.

Theorem 5. *Suppose c is a constant that is not used in $\Gamma, \psi(x)$ and φ. If $\Gamma, \psi(x \mapsto c) \vDash \varphi$, then we have $\Gamma, \exists x \psi(x) \vDash \varphi$; if $\Gamma \vDash \psi(x \mapsto c)$, then we have $\Gamma \vDash \forall x \psi(x)$.*

This can also be proved by interpretation, so that is left as exercise (*hint: prove by contradiction*).

3.5 Equivalence

The quantifiers create some new equivalence, which are important to build up the intuition about the first order logic.

Firstly, all the equivalence for propositional logic is also an equivalence for first-order logic. For example, we know that

$$\forall x \varphi(x) \to \psi \equiv \neg\forall x \varphi(x) \vee \psi.$$

Here we will not explicitly give all the equations coming from propositional logic.

One of the most important equivalence in first-order logic is about the negative before the quantifier. More preciously, we have

$$\neg\forall x \varphi(x) \equiv \exists x \neg\varphi(x), \tag{8}$$

$$\neg\exists x \varphi(x) \equiv \forall x \neg\varphi(x). \tag{9}$$

We will just prove the first identity, since the second one can be deduced with the first one and the double negative elimination. Suppose in the model $\mathcal{M}$, it is not the case such that all $c^{\mathcal{M}}$ satisfies $\varphi(x \mapsto c)$ is true, then there must be some $c'^{\mathcal{M}}$ such that $\varphi(x \mapsto c')$ is not true. Conversely, if there exists some $c^{\mathcal{M}}$ such that $\varphi(x \mapsto c)$ is not true, it is not the case such that all $c'^{\mathcal{M}}$ satisfies $\varphi(x \mapsto c')$. Again, a straightforward proof.

A special case of the equivalent which is used extensively to express a statement in first-order logic is given as follow.

$$\neg\forall x(\varphi(x) \to \psi(x)) \equiv \exists x(\varphi(x) \wedge \neg\psi(x)).$$

An example is, suppose $\rho(x)$ means that x is a person, $\varphi(x)$ means that x is died (with common meaning) and $\psi(x)$ means that x is alive (with special meaning, for example, he is remembered by people), then the sentence *some peaple have died, but he is still alive* can be expressed as

$$\exists x(\rho(x) \wedge \varphi(x) \wedge \psi(x)).$$

If it is not the case, we know that

$$\begin{aligned}&\neg\exists x(\rho(x) \wedge \varphi(x) \wedge \psi(x))\\ \equiv&\forall x((\rho(x) \wedge \varphi(x)) \to \neg\psi(x)).\end{aligned}$$

This is to say, everyone that is dead (with common meaning) is not alive (with special meaning), which agrees with our intuition.

The next one is the distributive law of the quantifiers with disjunction and conjuction.

$$\forall x(\varphi(x) \wedge \psi(x)) \equiv \forall x\varphi(x) \wedge \forall x\psi(x), \tag{10}$$

$$\exists x(\varphi(x) \vee \psi(x)) \equiv \exists x\varphi(x) \vee \exists x\psi(x). \tag{11}$$

This is actually straightforward, since $\forall$ can be considered to be a "big" conjunction and $\exists$ can be considered to be a "big" disjunction. If the domain only contains finite number of elements, a quantifier can be written be a series of $\wedge$ or $\vee$, in which this equation is trivial.

The third one is an important rule for the construction of prenex normal form for sentences. Suppose ψ is a sentence in which there is no free occurrence of x, then

$$\psi \to \forall x\varphi(x) \equiv \forall x(\psi \to \varphi(x)), \tag{12}$$

$$\psi \to \exists x\varphi(x) \equiv \exists x(\psi \to \varphi(x)). \tag{13}$$

The proof of the equivalence is left as exercise.

3.6 Prenex Normal Form

Just as the conjunctive and disjunctive normal form, we are interested in finding some simple representation of sentences in first order logic. A simple one of those representations is called prenex normal form of a sentence.

A sentence is called a prenex normal form, if all the quantifiers appears in front of the sentence. For example,

$$(\forall\epsilon)(\exists\delta)(\forall x)\varphi(x, \epsilon, \delta)$$

is in prenex normal form.

Theorem 6. *Each sentence in first-order logic has an equivalence prenex normal form.*

Proof. Since $\{\neg, \to\}$ is a complete set of conjuctions, we will firstly represent all the conjuctions with them to obtain an equivalent sentence. Then we rename all the variable, so that there are no variables bounded by different quantifiers to have the same name. Also we use double negative elimination to ensure that there is no double negative inside the sentence. Clearly, these operations will not change the sentence up to equivalence.

We will prove the result by induction on the *formulas*, i.e. we will show that for all formula φ, there exists a prenex normal form ψ such that they are equivalent if they agree on free variables. Suppose this is proved, then the result for sentence is trivial.

1. Suppose the formula contains no quantifier, i.e. an atomic proposition, then it is already in prenex normal form.
2. Suppose the formula is $\neg\varphi$. By induction hypothesis, φ has a prenex normal form φ'. We can move the negative inside the quantifiers with the equations in last section to obtain a prenex normal form.
3. Suppose the sentence is $\forall x\varphi(x)$ (or $\exists\varphi(x)$), by induction hypothesis, there exists $\psi(x)$ such that $\psi(x \mapsto c) \equiv \varphi(x \mapsto c)$ for all fixed c. It's easy to show that $\forall x\psi(x)$ (or $\exists x\psi(x)$) is the desired prenex normal form.
4. Suppose the sentence is $\varphi \to \psi$, by induction hypothesis we can find a prenex normal form $\psi' \equiv \psi$ and $\varphi' \equiv \varphi$. Then, with the last equivalence in the last section, we can replace φ by φ' and move it inside of the quantifiers. It's easy to see that we will obtain a prenex normal form which is equivalent to the original one. Note that renaming the variables helps us here.

□

Note that even if we are proving the property about *sentence*, we need to induction to the result about *formula*, since if we remove the quantifier of $\forall x\varphi(x)$, $\varphi(x)$ itself is not a sentence. It is actually a widely used trick to prove the result by strengthening the induction hypothesis.

To describe the arrange of quantifiers, we define the set of prenex normal forms starting with $\exists$ and have n times quantifier changes from left to right as Σ_n, and starting with $\forall$ as Π_n. For example,

$$(\forall\epsilon)(\exists\delta)(\forall x)\varphi(x,\epsilon,\delta) \in \Pi_3, \tag{14}$$

$$(\exists x)(\exists y)P(x,y) \in \Sigma_1. \tag{15}$$

4 Proof Systems

Until now we have developed two different languages for statements: propositional logic and first-order logic. It is possible to describe a statement use these languages, but now we do not mention what is the deduction, or more mathematically, the proof. If we can formally describe a system in which we can do logical deduction, it will be of great convenience for working mathematicians.

We will describe two proof systems, one is called the natural deduction system and another one is called the Hilbert-style system. The natural deduction system is introduced by Gentzen and the Hilbert-style system, just as its name, is introduced by Hilbert.

4.1 Natural Deduction: Propositional Logic

The basic idea of natural deduction is to characterize how to introduce and eliminate a logical symbol. A proof in natural deduction is a tree, in which the leaves are labeled by assumptions (maybe open, which will be described later) and the root is labeled by the conclusion.

We will start with the natural deduction system for propositional logic and then extend it to support first-order logic. Suppose Γ is a set of sentences which are considered as assumptions and φ is the conclusion we want. A valid proof tree must be with root φ and has no open assumptions, which will be introduced later. The proof tree is defined inductively.

Any sentence itself can form a tree (with only one node). Each sentence that is not in Γ should be marked with a label and is considered to be open, and all open assumptions forms the open assumption set of the proof tree.

Suppose there is two proof trees with root ψ and ρ, we can build a new tree with root $\psi \land \rho$ and the open assumption set of the new tree is the union of that of ψ and ρ. Conversely, suppose there is a proof tree with root $\psi \land \rho$, we can construct a new proof tree with root ψ (or ρ) and the assumption set of it remains unchanged. These two rules are called introduction and elimination rules of $\land$. We may represent this rule with

$$(\land I)\frac{\overset{\vdots}{\psi} \quad \overset{\vdots}{\rho}}{\psi \land \rho} \qquad (\land E)\frac{\overset{\vdots}{\psi \land \rho}}{\psi} \qquad \frac{\overset{\vdots}{\psi \land \rho}}{\rho}. \tag{16}$$

Similar, we can draw the introduction and elimination rule for $\lor$, as

$$(\lor I)\frac{\overset{\vdots}{\psi}}{\psi \lor \rho} \qquad \frac{\overset{\vdots}{\rho}}{\psi \lor \rho} \tag{17}$$

$$(\lor E)\frac{\overset{\vdots}{\psi \lor \rho} \quad \overset{\vdots}{\psi \to \theta} \quad \overset{\vdots}{\rho \to \theta}}{\theta}. \tag{18}$$

The introduction rule for $\lor$ is straightforward, so we will only explain the elimination rule of it. Suppose we have $\psi \lor \rho$, and both ψ and ρ will deduce θ, we can immediately know that θ is true.

The rules for $\to$ is much more interesting, since we can "close" those open assumptions by introducing $\to$. Suppose we have $\psi \to \rho$ and ψ, it is clear that we can deduce ρ, which is formally described by the elimination rule

$$(\to E)\frac{\overset{\vdots}{\psi \to \rho} \quad \overset{\vdots}{\psi}}{\rho}. \tag{19}$$

And if we have ρ with an *open* assumption ψ, which means that we construct a procedure to build a proof of ρ with assumption ρ. This is exactly the meaning of $\psi \to \rho$. As a result, the introduction rule is

$$(\to I)\frac{\begin{array}{c}[\psi]^i \\ \vdots \\ \rho\end{array}}{\psi \to \rho}, \tag{20}$$

Note that $[\psi]^i$ is closed in the new tree with root $\psi \to \rho$.

There is some technical trick to describe $\neg$ in the natural deduction system. We will define a symbol $\bot$, means a contradiction, and $\neg\varphi$ is defined as $\varphi \to \bot$. Clearly,

$\perp$ has no introduction rule, since we can not introduce a contradiction (and of course, the only way to proof $\perp$ is to use the elimination rule of $\to$). The elimination rule of $\perp$, usually called the double negative elimination, is

$$(DNE)\frac{\begin{matrix}\vdots\\ \neg\neg\varphi\end{matrix}}{\varphi}. \tag{21}$$

Clearly $\neg\neg\varphi \vDash \varphi$, so this rule is consistent. We can also introduce $\neg$ with $\perp$ as

$$(\neg I)\frac{\begin{matrix}\vdots\\ \varphi\to\perp\end{matrix}}{\neg\varphi} \quad (\neg E)\frac{\begin{matrix}\vdots\\ \neg\varphi\end{matrix} \quad \begin{matrix}\vdots\\ \varphi\end{matrix}}{\perp}. \tag{22}$$

If there is a proof tree with assumption Γ and root φ, φ is called provable from Γ, denoted as $\Gamma \vdash_{ND} \varphi$. If the proof system to use is clear in context, we can just use $\Gamma \vdash \varphi$. The following lemma states the relationship between $\vDash$ and $\vdash_{ND}$.

Lemma 3. *Suppose* $\Gamma \vdash_{ND} \varphi$, *then* $\Gamma \vDash \varphi$.

The proof can be done by structural induction on the proof tree, i.e. each proof tree with open assumptions Γ and root φ satisfies $\Gamma \vDash \varphi$, and the technical detail is omitted. In particular, if a set of sentences $\Gamma \vdash_{ND} \perp$, it is called inconsistent. It's not hard to know that an inconsistent set is unsatisfiable.

Conversely, a natural question is that whether it is true that $\Gamma \vDash \varphi$ implies $\Gamma \vdash_{ND} \varphi$. A proof system P is called complete, if $\Gamma \vDash \varphi$ implies $\Gamma \vdash_P \varphi$. In Section 5 we will prove that both natural deduction and Hilbert-style system are complete for propositional and first-order logic.

4.2 Examples about Negative

We will show an example of natural deduction system by proving some equivalence rules for $\neg$. Firstly, we will show that double negative elimination implies proof by contradiction, which is

$$(PBC)\frac{\begin{matrix}[\neg\varphi]\\ \vdots\\ \perp\end{matrix}}{\varphi}. \tag{23}$$

We can simulate such rule by double negative elimination as

$$\cfrac{\cfrac{\cfrac{\begin{matrix}[\neg\varphi]\\ \vdots\\ \perp\end{matrix}}{\neg\varphi\to\perp}}{\neg\neg\varphi}}{\varphi}.$$

Conversely, it is not hard to show DNE can be simulated by PBC. Thus contains any one of them can obtain the systems with equivalent proof ability.

Similarly, the law of excluded middle, described as

$$(EM)\frac{}{\varphi \vee \neg\varphi} \tag{24}$$

is equivalent to both DNE and PBC. The detail of proof is omitted.

Another weaker rule is called the law of explosion, described as

$$\frac{\begin{matrix}\vdots \\ \bot\end{matrix}}{\varphi}, \tag{25}$$

where φ is any sentence. This rule is reasonable since if we have any introduction in the assumption Γ, then of course $\Gamma \cup \{\neg\varphi\}$ is unsatisfiable for all φ, so that $\Gamma \vDash \varphi$. This rule can be proved by PBC as

$$\cfrac{\cfrac{\cfrac{\begin{matrix}\vdots \\ \bot\end{matrix} \quad [\varphi]}{\bot \wedge \varphi}}{\bot}}{\varphi}.$$

But conversely, the low of explosion *can not* imply both DNE and PBC. If we remove DNE/PBC and add the low of explosion, we will get a weaker deduction system called intuitionistic logic.

4.3 Natural Deduction: First-Order Logic

Now we are ready to support first-order logic in natural deduction system.

Suppose $\varphi(x)$ is a formula with only one free variable x, and c is a new constant that has no occurrence in $\varphi(x)$ and any assumptions, then we have

$$(\forall I)\frac{\begin{matrix}\vdots \\ \varphi(x \mapsto c)\end{matrix}}{\forall x\varphi(x)}. \tag{26}$$

This means that if we can prove $\varphi(x \mapsto c)$ for a new constant c (i.e. no special property), then we can prove it for all constant. Similarly, we can write the elimination rule for $\forall$ as

$$(\forall E)\frac{\begin{matrix}\vdots \\ \forall x\varphi(x)\end{matrix}}{\varphi(x \mapsto t)}, \tag{27}$$

where t is any term without variable. The elimination rule is actually straightforward. As we have mentioned, $\forall$ is just like a “big $\wedge$”, so their introduction/elimination rules are similar.

Suppose $\varphi(x)$ is some formula with only one free variable x, t is any term without variable and c is a constant that has no occurrence in $\varphi(x)$, then we have

$$(\exists I)\frac{\begin{array}{c}\vdots\\ \varphi(x \mapsto t)\end{array}}{\exists x \varphi(x)} \tag{28}$$

$$(\exists E)\frac{\begin{array}{c}\vdots\\ \exists x \varphi(x)\end{array} \qquad \begin{array}{c}\vdots\\ \varphi(x \mapsto c) \to \psi\end{array}}{\psi}. \tag{29}$$

Again, the elimination rule is quite similar to that of $\vee$.

To define the identity $=$, we also need the introduction rules and elimination rules, which is defined as

$$(=_1)\frac{}{t = t} \tag{30}$$

$$(=_2)\frac{s = t \qquad \varphi(x \mapsto s)}{\varphi(x \mapsto t)} \tag{31}$$

$$\frac{t = s \qquad \varphi(x \mapsto s)}{\varphi(x \mapsto t)}, \tag{32}$$

which is fairly straightforward.

Since we have introduced tons of rules in this section, it is necessary to do some exercise to get familiar with them. We will prove one side of the equivalence that we have used in Section 3.6,

$$\begin{gathered}\psi \to \forall x \varphi(x) \vdash \forall x(\psi \to \varphi(x)),\\ \psi \to \exists x \varphi(x) \vdash \exists x(\psi \to \varphi(x)).\end{gathered}$$

A possible proof tree for the first example is

$$\dfrac{\dfrac{\dfrac{\dfrac{\psi \to \forall x \varphi(x) \qquad [\psi]_1}{\forall x \varphi(x)}}{\varphi(x \mapsto c)}}{\psi \to \varphi(x \mapsto c)^\star}}{\forall x(\psi \to \varphi(x))}$$

Note that the open assumption $[\psi]_1$ is closed at the position we marked by $\star$. A possible proof tree for the second example, which is much more non-trivial, uses the law of excluded middle.

$$\frac{\psi \vee \neg\psi \quad \begin{array}{c}\vdots\\ \psi \to \exists x(\psi \mapsto \varphi(x))\end{array} \quad \begin{array}{c}\vdots\\ \neg\psi \to \exists x(\psi \mapsto \varphi(x))\end{array}}{\exists x(\psi \mapsto \varphi(x))}.$$

The detail of the proof is left as exercise.

4.4 Hilbert-Style System: Propositional Logic

Natural deduction system, at least after some practice, is fairly natural for humans to do reasoning, but it is sometimes too heavy for computers or in mathematical proofs. It contains a lot of rules, so that if we want to prove some property of the natural deduction system, we have to do a long intuition over all rules. We may be satisfied if there exists a simple system which is "equivalent" to the natural deduction system.

Hilbert-style system is one of the candidates of the simpler systems. Suppose Γ is a set of sentences which are considered as assumption and φ is the target of the proof. A proof in Hilbert-system (H for short) is a sequence of statements ended with φ, and each line in the sequence is

1. an axiom, which comes from a finite number of axiom schemes; or
2. the application of a rule, where there is only one rule for propositional logic and two for first-order logic; or
3. an assumption, i.e. a sentence in Γ.

For propositional logic, there are three axiom schemes in H.

Axiom 1 $\varphi \to \psi \to \varphi$;

Axiom 2 $(\varphi \to \psi \to \rho) \to (\varphi \to \psi) \to \varphi \to \rho$;

Axiom 3 $\neg\neg\varphi \to \varphi$.

Each axiom scheme can form infinite number of axioms, by replacing the variables φ, ψ and ρ with any formulas. For technical convenience, we may regard $\neg\varphi$ as an alias of $\varphi \to \bot$, and other connectives like $\wedge, \vee$ are defined by $\neg$ and $\to$, since $\{\neg, \to\}$ is a complete set of connectives.

The only rule for propositional logic is called Modus Ponens, described as

$$(MP)\frac{\varphi \to \psi \quad \varphi}{\psi}. \tag{33}$$

More preciously, if both $\varphi \to \psi$ and φ occurs the sequence of proof before, we can add ψ to the end of the proof.

As usual, we define $\Gamma \vdash_H \varphi$ if φ is provable in H with assumptions Γ. In particular, if φ is provable from H without any assumption, we will say $\vdash_H \varphi$, where we can use $\vdash$ if there is no ambiguity. A set of sentences Γ is called inconsistent, if $\Gamma \vdash \bot$. If Γ is inconsistent, then for all sentence φ, we know that $\Gamma \vdash \varphi$, and the proof with assumption $\bot$ is given as follow.

Id	Proof	Reason
1	$\bot$	Assumption
2	$\bot \to \neg\varphi \to \bot$	Axiom 1
3	$\neg\neg\varphi$	MP from 1,2
4	$\neg\neg\varphi \to \varphi$	Axiom 3
5	φ	MP from 3,4

Table 3: $\bot \vdash \varphi$.

We may use a very unnatural but critical example for the proof in H, that is $\varphi \to \varphi$ for all fomula φ. This is an important building block for the deduction theorem which connects H with natural deduction system. We define

$$A = (\varphi \to (\varphi \to \varphi) \to \varphi)$$
$$B = \varphi \to \varphi \to \varphi,$$

and the proof is given in the following table.

Id	Proof	Reason
1	A	Axiom 1
2	$A \to B \to \varphi \to \varphi$	Axiom 2
3	$B \to \varphi \to \varphi$	MP from 1,3
4	B	Axiom 1
5	$\varphi \to \varphi$	MP from 3,4

Table 4: $\varphi \to \varphi$.

4.5 Hilbert-Style System: First-order logic

Now we will extend system H to first-order logic. There are six more axiom schemes for first-order logic, where the first four are for quantifiers.

Axiom 4 $\forall x \varphi(x) \to \varphi(t)$;

Axiom 5 $\forall x(\psi \to \varphi(x)) \to \psi \to \forall x \varphi(x)$;

Axiom 6 $\varphi(t) \to \exists x \varphi(x)$;

Axiom 7 $\forall x(\varphi(x) \to \psi) \to \exists x \varphi(x) \to \psi$,

where $\varphi(x)$ is a formula with no free variable except x, and ψ is a sentence. And another rule called for the universal quantifier is introduced

$$(R\forall)\frac{\varphi(x \mapsto c)}{\forall x \varphi(x)}, \tag{34}$$

in which c has no occurrence in $\varphi(x)$ or any assumptions in Γ. This is to say, if $\varphi(x \mapsto c)$ is true for some *new* c that is not occurred in assumptions and $\varphi(x)$ itself, then $\varphi(x)$ is universally correct.

The last two axiom schemes are for the identity $=$.

Axiom 8 $\forall x\ x = x$.

Axiom 9 $\forall x \forall y(x = y \to \varphi(x) \to \varphi(y))$.

Note that each axiom scheme can generate infinite number of axioms, by replacing $\varphi(x)$ with any formula with at most free variable x and replace ψ with any sentence, where x can be arbitrary variable.

4.6 Deduction Theorem

Deduction theorem gives us a systematical way to find a proof in Hilbert-style system. It is similar to one of the properties of logical consequence introduced in Section 2.7.

Theorem 7 (Deduction Theorem)**.** *Suppose Γ is a set of sentences and φ is a sentence, $\Gamma \vdash \varphi \to \psi$ if and only if $\Gamma, \varphi \vdash \psi$.*

Proof. Suppose $\Gamma \vdash \varphi \to \psi$, from the assumption $\Gamma \cup \{\varphi\}$, we can prove ψ in the following way.

Id	Proof	Reason
1	$\varphi \to \psi$	$\Gamma \vdash \varphi \to \psi$
2	φ	Assumption
3	ψ	MP from 1,2

By the definition, we know that $\Gamma, \varphi \vdash \psi$.

The other side of the proof is a bit more complex. Suppose $\Gamma, \varphi \vdash \psi$, by the definition, there exists a proof (i.e. sequence of sentence) in H from $\Gamma \cup \varphi$ to ψ. Let the proof be $(\rho_1, \rho_2, \dots, \rho_n)$, we will *induction* on n, to show that we can obtain a proof $\varphi \to \rho_i$ from assumption Γ for all $1 \leq i \leq n$. In other words, we remove the φ from the assumption and add it in front of each line of the original proof.

Suppose $n = 1$, then the only sentence in the proof is ψ, so that it must be an assumption in $\Gamma \cup \varphi$. If $\psi = \varphi$, we just need to prove $\varphi \to \varphi$, which has already been done; otherwise is $\psi \in \Gamma$, we can give the following proof.

Id	Proof	Reason
1	ψ	Assumption
2	$\psi \to \varphi \to \psi$	Axiom 1
3	$\varphi \to \psi$	MP from 1,2

Now assume $n > 1$, and by induction hypothesis, we have already proved

$$\begin{aligned} \rho_1' &= \varphi \to \rho_1, \\ \rho_2' &= \varphi \to \rho_2, \\ &\dots \\ \rho_n' &= \varphi \to \rho_1. \end{aligned}$$

Consider the last sentence ρ_n in the original proof. If ρ_n is an assumption, it is just like the case for $n = 1$. If ρ_n is an axiom, since we can get ψ from nothing, we can prove $\varphi \to \psi$ use the method for $n = 1$.

If ρ_n is obtained by Modus Ponens, i.e. there exists $i, j < n$, such that

$$\begin{aligned} \rho_i &= \alpha \to \rho_n \\ \rho_j &= \alpha, \end{aligned}$$

then by the induction hypothesis, after removing the assumption φ we can prove

$$\begin{aligned} \rho_i' &= \varphi \to \alpha \to \rho_n \\ \rho_j' &= \varphi \to \alpha. \end{aligned}$$

We can use axiom 2 to prove $\varphi \to \psi$.

Id	Proof	Reason
1	$\rho_i' \to \rho_j' \to \varphi \to \rho_n$	Axiom 2
2	$\rho_j' \to \varphi \to \rho_n$	MP from φ_i' and 1
3	$\varphi \to \rho_n$	MP from φ_j' and 2

In the last case, if ρ_n is obtained by $\forall$ introduction rule, i.e. there exists some $i < n$ such that $\rho_i = \varphi(x \mapsto c)$ in the original proof. By the induction hypothesis, after removing φ from the assumption, we can get $\psi \to \varphi(x \mapsto c)$. Then we can use axiom 5 to obtain the proof we want.

Id	Proof	Reason
1	$\psi \to \varphi(x \mapsto c)$	Induction
2	$\forall x(\psi \to \varphi(x))$	$\forall$ Rule
3	$\forall x(\psi \to \varphi(x)) \to \psi \to \forall x \varphi(x)$	Axiom 5
4	$\psi \to \forall x \varphi(x)$	MP from 2,3

In conclusion, we described a process to transform the proof of $\Gamma, \varphi \vdash \psi$ to the proof of $\Gamma \vdash \varphi \to \psi$, so that the deduction theorem holds for both sides. $\square$

5 Soundness and Completeness

5.1 Preparations

We have already known the semantics and proof system for propositional logic and first-order logic, and in this section, we will connect *logical consequence* (defined by semantic meaning) with *formal deduction* (defined in proof systems) by the following theorem.

Theorem 8. *Suppose Γ is a set of sentence and φ is a sentence in first-order logic, then the following conditions are equivalent.*

1. $\Gamma \vDash \varphi$.
2. $\Gamma \vdash_H \varphi$;
3. $\Gamma \vdash_{ND} \varphi$;

The property that $\Gamma \vDash \varphi$ implies $\Gamma \vdash \varphi$ is called the completeness of the proof system, which means that our proof systems have "complete" ability in proving. The property that $\Gamma \vdash \varphi$ implies $\Gamma \vDash \varphi$ is called the soundness of the proof system, which means that the proof in the system is reasonable.

Recall that

1. $\Gamma \vDash \varphi$ if and only $\Gamma \cup \{\varphi\}$ is unsatisfiable, i.e. there does not exists a model in which all those sentences are satisfied;
2. $\Gamma \vdash_{ND} \varphi$ if and only if $\Gamma \cup \{\varphi\}$ is inconsistent in natural deduction system, i.e. the assumptions deduce $\perp$;
3. $\Gamma \vdash_H \varphi$ if and only if $\Gamma \cup \{\varphi\}$ is inconsistent in Hilbert-style system, i.e. the assumptions deduce $\perp$.

By the law of excluded middle, we know that for both natural deduction and Hilbert-style system, $\Gamma \vdash \varphi$ if and only if $\Gamma, \neg\varphi \vdash \perp$, therefore if the following lemma holds, we can immediately deduce Theorem 8.

Lemma 4. *Suppose Γ is a set of sentence in first-order logic, then the following conditions are equivalent.*

1. *Γ is satisfiable;*

2. Γ is consistent in natural deduction system;

3. Γ is consistent in Hilbert-style system.

Our proof contains three parts: (1) $\Rightarrow$ (2), (2) $\Rightarrow$ (3) and (3) $\Rightarrow$ (1).

5.2 Part 1: (1) $\Rightarrow$ (2)

In the first part we will show that if Γ is satisfiable, it is also consistent in natural deduction system. This is equivalent to the following statement: if Γ is inconsistent in natural deduction system, it must be unsatisfiable. Then our goal is to show that $\Gamma \vdash_{ND} \bot$ implies the unsatisfiability of Γ. We will use the following lemma to show this.

Lemma 5. *Suppose $\Gamma \vdash_{ND} \varphi$, then for any model $\mathcal{M}$ in which Γ is satisfied, φ is also satisfied. In other words, $\mathcal{M} \vDash \Gamma$ implies $\mathcal{M} \vDash \varphi$.*

Since we need to induction on the proof trees, we generalize the result for the proof trees with open assumptions in the following lemma.

Lemma 6. *Suppose there exists a proof tree for φ from assumptions Γ, containing open assumptions Γ', then for any model $\mathcal{M}$, $\mathcal{M} \vDash \Gamma \cup \Gamma'$ implies $\mathcal{M} \vDash \varphi$.*

If Lemma 6 is correct, we can also show Lemma 5 by considering the case where $\Gamma' = \varnothing$. Moreover, the case where $\varphi = \bot$ gives the proof of the original statement.

Note that $\bot$ is not introduced in the semantic interpretation of first-order logic. In fact, it can be any contradictory sentence, for example, $\varphi \to \neg\varphi$. For simplicity, we will ignore such technical details in the proof. Also, we can just consider the connectives $\to$ and $\bot$ in natural deduction system.

The proof of 6 is given by induction on the proof tree. There are a lot of cases to deal with, among which we only want to mention two important cases as examples.

The first case is that, the proof tree is obtained by the introduction rule of $\to$, i.e.

$$\frac{\begin{array}{c}[\psi]^i \\ \vdots \\ \rho\end{array}}{\psi \to \rho}.$$

By induction hypothesis, we know that $\mathcal{M} \vDash \Gamma \cup \{\psi\} \cup A$ implies $\mathcal{M} \vDash \rho$, where A is the set of open assumptions for the proof tree. If $\mathcal{M} \vDash \psi$, clearly $\mathcal{M} \vDash \Gamma \cup A$ implies $\mathcal{M} \vDash \Gamma \cup \{\psi\} \cup A$, so that it also imples $\psi \to \rho$; otherwise if $\mathcal{M} \vDash \neg\psi$, of course we have $\mathcal{M} \vDash \psi \to \rho$. This means that $\mathcal{M} \vDash \Gamma \cup A$ always implies $\psi \to \rho$.

The second case is that, the proof tree is obtained by the introduction rule of $\forall$, i.e.

$$(\forall I)\frac{\begin{array}{c}\vdots \\ \varphi(x \mapsto c)\end{array}}{\forall x \varphi(x)}.$$

By induction hypothesis, we know that $\mathcal{M} \vDash \Gamma \cup A$ implies $\mathcal{M} \vDash \varphi(x \mapsto c)$. Suppose there exists a model satisfying $\Gamma \cup A$ in which $\forall x \varphi(x)$ is not true, there must be some $c'^{\mathcal{M}}$ such that $\varphi(x \mapsto c')$ is not satisfied. Then we can easily construct a model in which $\varphi(x \to c)$ is not true, by interpreting $c^{\mathcal{M}}$ as $c'^{\mathcal{M}}$ in the previous model. As a result, $\mathcal{M} \vDash \Gamma \cup A$ always implies $\forall x \varphi(x)$.

The remaining part of the proof is omitted.

5.3 Part 2: (2) $\Rightarrow$ (3)

In the second part, we need to show that (2) $\Rightarrow$ (3), which is equivalent to the following statement: if Γ is inconsistent in Hilbert-style system, it is also inconsistent in natural deduction system. Similar to the first part, we will prove a more general result, and $\varphi = \perp$ deduces the result we want.

Lemma 7. *Suppose Γ is a set of sentences and φ is a sentence, then $\Gamma \vdash_H \varphi$ implies $\Gamma \vdash_{ND} \varphi$.*

For a sequence of sentences $(\varphi_1, \varphi_2, \ldots, \varphi_n)$ which is a proof in Hilbert-style system, we need to show that φ_i is also provable in natural deduction system, for all $1 \leq i \leq n$. This could be done by induction on the length of the proof, like we what we did in Section 4.6.

If φ_n is an assumption, it is obviously provable in natural deduction system; otherwise if it is an axiom, the following trees in the figure show that axioms in Hilbert-style system are provable in natural deduction system.

$$\dfrac{\dfrac{\dfrac{[\varphi]_1 \quad [\psi]_2}{\varphi \land \psi}}{\varphi}}{\varphi \to \psi \to \varphi} \tag{35}$$

$$\dfrac{\dfrac{\dfrac{[\varphi \to \psi \to \rho]_1 \quad [\varphi]_2}{\psi \to \rho} \qquad \dfrac{[\varphi \to \psi]_3 \quad [\varphi]_2}{\psi}}{\rho}}{(\varphi \to \psi \to \rho) \to (\varphi \to \psi) \to \varphi \to \psi} \tag{36}$$

$$\dfrac{\dfrac{[\neg\neg\varphi]}{\varphi}}{\neg\neg\varphi \to \varphi} \tag{37}$$

$$\dfrac{\dfrac{[\forall x \varphi(x)]_1}{\varphi(t)}}{\forall x \varphi(x) \to \varphi(t)} \tag{38}$$

$$\dfrac{\dfrac{\dfrac{\dfrac{[\forall x(\psi \to \varphi(x))]_1}{\psi \to \varphi(c)} \quad [\psi]_2}{\varphi(c)}}{\forall x \varphi(x)}}{\forall x(\psi \to \varphi(x)) \to \psi \to \forall x \varphi(x)} \tag{39}$$

$$\dfrac{\dfrac{[\varphi(t)]_1}{\exists x \varphi(x)}}{\varphi(t) \to \exists x \varphi(x)} \tag{40}$$

$$\dfrac{\dfrac{\dfrac{[\forall x(\varphi(x) \to \psi)]_1}{\varphi(t) \to \psi} \qquad \dfrac{[\exists x \varphi(x)]_2}{\varphi(t)}}{\psi}}{\forall x(\varphi(x) \to \psi) \to \exists x \varphi(x) \to \psi} \tag{41}$$

$$\dfrac{c = c}{\forall x\ x = x} \tag{42}$$

$$\dfrac{\dfrac{\dfrac{[c = d]_1 \quad \varphi(c)}{\varphi(d)}}{c = d \to \varphi(c) \to \varphi(d)}}{\forall x \forall y(x = y \to \varphi(x) \to \varphi(y))} \tag{43}$$

Suppose φ_n is neither an assumption nor an axiom, it must be obtained by a rule. If φ_n is obtained by Modus Ponens, i.e. there exist $1 \leq i, j < n$ such that

$$\begin{aligned} \varphi_i &= \alpha \to \varphi_n \\ \varphi_j &= \alpha. \end{aligned}$$

By induction hypothesis, α and $\alpha \to \varphi_n$ are provable, then we can construct the following proof tree of φ_n.

$$\dfrac{\begin{matrix} \vdots \\ \alpha \to \varphi_n \end{matrix} \qquad \begin{matrix} \vdots \\ \alpha \end{matrix}}{\varphi_n} \tag{44}$$

Otherwise if φ_n is obtained by the introduction rule of $\forall$, it can also be proved by the introduction rule of $\forall$ in natural deduction system.

The second part of the proof also gives a process to transform a proof in Hilbert-style system into a proof in natural deduction system.

5.4 Part 3(1): $(*) \Rightarrow (1)$

The third part of the proof is of great importance, since in the first two parts we are transforming the proof from a "simpler" system into a more "complex" system. In the third part, we need to prove the following theorem called model existence theorem.

Theorem 9. *Suppose Γ is a set of sentences which is consistent in Hilbert-style system, it is also satisfiable, i.e. there exists a model $\mathcal{M}$ such that $\mathcal{M} \vDash \Gamma$.*

The proof contains two steps:

1. we show that a maximal and $\exists$ complete consistent set of sentences have a model, and
2. we show that any consistent set can be extended to a maximal and $\exists$ complete consistent set of sentences.

Suppose L is the vocabulary, a set of sentences with vocabulary L denoted as Δ is called maximal, if for all sentence $\varphi \notin \Delta$, $\Delta \cup \{\varphi\}$ is inconsistent (in system H). Δ is called $\exists$ complete, if for all $\exists x\varphi(x) \in \Delta$, there exists some $c \in L$ such that $\varphi(x \mapsto c) \in \Delta$.

Lemma 8. *Suppose Δ is a consistent set of sentences from vocabulary L, if it is both maximal and $\exists$ complete, it must have a model.*

Proof. Define $\sim$ is a binary relation between the terms in L, where $a \sim b$ if and only if $a = b \in \Delta$. We will show that $\sim$ is an *equivalent relation*, in other words,

1. $a \sim a$, this is because $a \sim a$ is a tautology. If $a = a \notin \Delta$, $\Delta \cup \{a = a\}$ is still consistent, which contradicts to the fact that Δ is maximal.
2. $a \sim b$ and $b \sim c$ implies $a \sim c$, this is also because the maximality of Δ.
3. $a \sim b$ imples $b \sim a$, also because of maximality of Δ.

Since $\sim$ is a equivalent relation, it induces a partition of the terms in L, where $c \in L$ belongs to the block $[c]$.

Let C be the set of constants in the vocabulary, clearly each one of them occurs in Δ. We will construct a model in which the terms are interpreted as the block it belongs to, i.e. $t^{\mathcal{M}} = [t]$. For any function f with parameters $t_1, t_2, \dots, t_n$, suppose there exists some $[c_m]$ such that

$$f(t_1, t_2, \dots, t_n) = c_m \in \Delta,$$

we can define

$$f^{\mathcal{M}}([t_1], [t_2], \dots, [t_n]) = [c_m].$$

Note that by carefully checking (a structure induction), we know that f and the interpretation of terms are well-defined (i.e. the value of f does not depend on the choice of representation t_i), since Δ is consistent and maximal. The detail of the proof is omitted.

Now we have given the interpretation of all terms, and the next step is to give the interpretation of atomic propositions. For any predicates symbol P (including the identity) with n parameters, $P^{\mathcal{M}}(t_1^{\mathcal{M}}, \dots, t_n^{\mathcal{M}})$ if and only if

$$P(t_1, \dots, t_n) \in \Delta.$$

Such definition is well-defined since the interpretation for terms is well-defined.

After definition such model $\mathcal{M}$, we need to check that $\mathcal{M} \vDash \Delta$. Again since we need to prove the result by structure induction over the definition of formulas, we need to generalize the the result for formula, that is, for all formula $\varphi(x_1, \dots, x_n)$ with free variable $x_1, \dots, x_n$, and all $t_1, \dots, t_n \in \mathcal{M}$, $\varphi(x_1 \mapsto t_1, \dots, x_n \mapsto t_n) \in \Delta$ if and only if $\mathcal{M} \vDash \varphi(x_1 \mapsto t_1, \dots, x_n \mapsto t_n)$.

Suppose the formula is an atomic formula, by the definition of $P^{\mathcal{M}}$ we immediately have the result. And the remaining part of the induction is given by the propositions.

1. $\varphi \in \Delta$ if and only if $\neg\varphi \notin \Delta$.
2. $\Delta \vdash_H \varphi$ deduces that $\varphi \in \Delta$.
3. $\varphi \to \psi \in \Delta$ if and only if $\varphi \notin \Delta$ or $\psi \in \Delta$.
4. $\forall x\varphi(x) \in \Delta$ if and only if for all constant c, $\varphi(x \mapsto c) \in \Delta$.
5. $\exists x\varphi(x) \in \Delta$ if and only if there exists some constant c such that $\varphi(x \mapsto c) \in \Delta$.

The proof of these propositions is omitted (mainly uses the maximality and $\exists$ consistency of Δ). With these propositions, the induction is fairly straightforward. Note that this part of proof directly rely on the proving ability of system H. $\square$

5.5 Part 3(2): (3) $\Rightarrow$ (*)

The last step of the whole proof is to show the Lindenbaum's Lemma.

Lemma 9. *Suppose L is some vocabulary of first-order logic, and Δ is a consistent set of sentences. Let $L' = L \cup \{c_0, c_1, \dots\}$ for new variables $c_0, c_1, \dots$, there exists a maximal consistent and $\exists$ complete set of sentences Δ' containing Δ.*

Proof. Clearly each sentence with vocabulary L' can be encoded into binary string with finite length, and then be encoded into natural numbers. This means that we can enumerate all the sentences one by one as

$$\varphi_1, \varphi_2, \dots, \varphi_n, \dots$$

Let $\Delta_0 = \Delta$. For $i = 1, 2, \dots$, we can construct Δ_i from Δ_{i-1} by the following algorithm.

1. If $\Delta_{i-1} \cup \{\varphi_i\}$ is inconsistent, $\Delta_i = \Delta_{i-1}$.
2. Otherwise if φ_i is not existentially quantified, $\Delta_i = \Delta_{i-1} \cup \{\varphi_i\}$.
3. Otherwise if $\varphi_i = \exists x\psi(x)$, we have $\Delta_i = \Delta_{i-1} \cup \{\varphi_i, \psi(c_k)\}$, for some c_k that is not used in $\Delta_0, \dots, \Delta_{i-1}$.

Clearly Δ_i is well-defined for all natural number i, let Δ^* be the union of all Δ_i.

By induction, it's not hard to show that

1. $\Delta_0 \subseteq \Delta_1 \subseteq \Delta_2 \subseteq \dots$
2. Δ_i is consistent for all $i \in \mathbb{N}$.

Suppose Δ^* is *inconsistent*, it can deduce $\bot$ in system H. Since in the deducing process only finite number of sentences are used, there exists some Δ_k such that $\Delta_k \vdash_H \bot$, which leads to a contradiction. As a result, Δ^* is consistent. Moreover, by the definition of the algorithm we know that it must be maximal and $\exists$ complete.

In conclusion, we construct a maximal consistent and $\exists$ complete set containing $\Delta_0 = \Delta$. $\square$

Theorem 8 gives very strong property of natural deduction system and Hilbert-style system.

Firstly, it shows that both natural deduction system and Hilbert-style system are sound, i.e. the deduction in such systems is logically reasonable.

Second, it shows that both of the systems are complete, i.e. they are sufficient for all kinds of logical deduction in first-order logic. In particular, the deduction ability of them is equivalent. Note that the soundness is one of the basic requirement of proof systems (not only for first-order logic), but completeness is not. In more expressive logic systems, the completeness may not hold.

Finally, it explicitly gives an algorithm to translate a proof from natural deduction system into Hilbert-style system, and provides a systematical way to construct a model from a consistent set of sentences. Actually, there also exists an algorithm to translate a proof in system H back to the natural deduction system by the deduction theorem, which will not be covered by this article.

6 Model Theory

6.1 Isomorphism of Model

In this section we will introduce some basic concepts and results about models.

Suppose L is some vocabulary, two models $\mathcal{M}$ and $\mathcal{M}'$ under the vocabulary L are called isomorphic to each other, denoted as $\mathcal{M} \cong \mathcal{M}'$, if they have the same structure up to permutation of *elements*. More preciously, there exists some bijection f from the element set of $\mathcal{M}$ to the element set of $\mathcal{M}'$, such that

1. for all constant $c \in L$, $f(c^{\mathcal{M}}) = c^{\mathcal{M}'}$.
2. for all predicates $P \in L$ and elements $a_1, \dots, a_n \in M$, $P^{\mathcal{M}}(a_1, \dots, a_n)$ if and only if $P^{\mathcal{M}'}(f(a_1), \dots, f(a_n))$; and
3. for all functions $g \in L$ and elements $a_1, \dots, a_n \in M$, $g^{\mathcal{M}}(a_1, \dots, a_n)$ if and only if $P^{\mathcal{M}'}(f(a_1), \dots, f(a_n))$.

The following lemma shows that isomorphic models are indistinguishable with sentences.

Lemma 10. *Suppose $\mathcal{M} \cong \mathcal{M}'$, then for any sentence φ, $\mathcal{M} \vDash \varphi$ if and only if $\mathcal{M}' \vDash \varphi$.*

Again, we need to generalize the result to formulas in order to prove it inductively.

Lemma 11. *Suppose $\mathcal{M} \cong \mathcal{M}'$, then for any formula $\varphi(x_1, \dots, x_n)$ with n free variables, for any $c_1, c_2, \dots, c_n \in L$, $\mathcal{M} \vDash \varphi(x \mapsto c)$ if and only if $\mathcal{M}' \vDash \varphi(x \mapsto c)$.*

The detail of the inductive proof is omitted, which is fairly straightforward.

Suppose T is a first-order theory (i.e. a set of sentences), there may be many models $\mathcal{M}$ in which T is satisfied. Furthermore, there may be several non-isomorphic models for a theory. For example, let T be the theory of equivalence relation, which contains the following axioms,

1. Reflexity: $\forall x R(x, x)$.
2. Transitivity: $\forall x \forall y (R(x, y) \to R(y, z) \to R(x, z))$.
3. Symmetry: $\forall x \forall y (R(x, y) \to R(y, x))$.

There are infinitely many models for this theory. For example, natural numbers with equality, $\mathbb{Z}/n\mathbb{Z}$ with equality, etc.

6.2 Consistency and Completeness of Theory

Suppose T is a theory, we can define $T \vdash \varphi$ for some sentence φ, if $T \vdash_H \varphi$. In such cases, φ is called a theorem of T.

A theory is called consistent, if $T \nvdash \bot$. By the law of explosion, an inconsistent theory deduces all sentences, so that it is not meaningful. By this observation, an equivalent definition of consistency is that there is some sentence that is not a theorem of T.

In fact, during the proof and completeness and soundness of first-order logic, we have already shown an important result (Theorem 9) about theory, that is a theory is consistent if and only if there is a model for it.

A theory T over vocabulary L is called complete, if it decides every sentence in L, i.e. either $T \vdash \varphi$ or $T \vdash \neg\varphi$ for any sentence φ.

A theory T is called categorical, if there is only one model for T up to isomorphism. Similarly, a theory T is called countably categorical, if all its countably infinite models are isomorphic. Note that a model is called countable, if set of the symbols (containing elements, relations and functions) is countable. Clearly a categorical theory must be complete, but conversely, it's not hard to find a complete theory that is not categorical.

Two models $\mathcal{M}$ and $\mathcal{M}'$ with the same vocabulary L are called elementarily equivalent, denoted as $\mathcal{M} \equiv \mathcal{M}'$, if they agrees on all sentences φ with vocabulary L, i.e. $\mathcal{M} \vDash \varphi$ if and only if $\mathcal{M}' \vDash \varphi$. In particular, if two models are isomorphic, they must be elementarily equivalent.

Note that elementarily equivalent models may have different structure, i.e. they may not be isomorphic. For example, sometimes we can extend a model by introducing some new elements. We will explicitly give an example of such method by constructing a nonstandard model for natural number which is elementarily equivalent to the standard model.

Lemma 12. *A theory T is complete if and only if all models of it are elementarily equivalent.*

Proof. Firstly if T is inconsistent, i.e. there is no model for it, this condition must hold.

Suppose T is complete, it decides every sentence φ. If there exists two models $\mathcal{M}, \mathcal{M}'$ of it such that $\mathcal{M} \vDash \psi$ and $\mathcal{M}' \nvDash \psi$, ψ can neither be true nor be false, which leads to a contradiction.

Conversely, suppose all models of it are elementarily equivalent, we can arbitrary choose a model $\mathcal{M}$ of it to decide the sentence φ.

1. If $\mathcal{M} \vDash \varphi$, all models of T agrees on it, so that $T \cup \{\neg\varphi\}$ is unsatisfiable, thus inconsistent. By the law of proving by contradiction, we know that $T \vdash \varphi$.
2. Otherwise if $M \vDash \neg\varphi$, since we have double negative elimination, the argument is similar.

□

6.3 Model $\mathcal{N}$

Now we will describe the standard model of natural numbers $\mathcal{N}$. Let $L = \{0,', +, \cdot\}$ be the vocabulary of first-order arithmetic, where $'$ is a unary function, $+$ and $\cdot$ are binary functions. For simplicity, $'(x)$, $+(x, y)$ and $\cdot(x, y)$ are denoted by x', $x + y$ and $x \cdot y$, respectively. In model $\mathcal{N}$, $+$ and $\cdot$ are addition and multiplication in natural number, and x' denotes $x + 1$. More formally, we can inductively define these operators by

1. Addition rules: $0 + x = x + 0 = x$, $x' + y = (x + y)'$.
2. Multiplication rules: $0 \cdot x = x \cdot 0 = 0$, $x' \cdot y = x \cdot y + y$.

Note that $\mathcal{N}$ is a model, and the interpretation of the model is obvious.

Suppose $\mathcal{M}$ is a model under vocabulary L, the theory of $\mathcal{M}$, denoted as $Th(\mathcal{M})$, is the set of all sentences φ such that $\mathcal{M} \vDash \varphi$.

Lemma 13. *For any model $\mathcal{M}$, $Th(\mathcal{M})$ is a maximal consistent set.*

Proof. Since $Th(\mathcal{M})$ exists a model, by the model existence theorem it must be consistent. Moreover, for all $\varphi \notin Th(\mathcal{M})$, by definition we know that $\mathcal{M} \nvDash \varphi$, so that $\mathcal{M} \vDash \neg\varphi$, which means that $\mathcal{M} \cup \{\varphi\}$ is unsatisfiable. By the model existence theorem, $M \cup \{\varphi\}$ is inconsistent, so that $\mathcal{M}$ must be maximal. □

By the lemma above, we know that $Th(\mathcal{N})$ is a maximal consistent set containing all the sentences that all correct in model $\mathcal{N}$. Clearly we know very little about $Th(\mathcal{N})$, for example, by now we don't know *whether there exists an integer larger than two which can not be written as the sum of three primes*, which is quite simple to define in first-order logic.

6.4 Compactness Theorem

Now we would like to introduce one of the most significant result in model theory of first-order logic, which gives a criteria to decide whether a theory is satisfiable. With the completeness theorem, we can obtain a very simple proof of it.

Theorem 10 (Compactness Theorem)**.** *Suppose Δ is a set of sentences, then Δ is satisfiable if and only if every finite subset of it is satisfiable.*

Proof. One side of the proof is straightforward. Suppose Δ is satisfiable, then every finite subset of it satisfies the model of Δ, thus it is satisfiable.

Conversely, suppose every finite subset of Δ is satisfiable, but Δ itself is unsatisfiable. By the completeness theorem we know that Δ is inconsistent, i.e. $\Delta \vdash \perp$. During the derivation in system H, only finite number of the sentences in Δ are used. Let Δ' be the set of sentences in Δ that is used in the derivation of $\Delta \vdash \perp$, then it is a finite inconsistent subset, thus a finite unsatisfiable subset, which leads to a contradiction. □

An application of the compactness theorem is to show that there exists a nonstandard model for $Th(\mathcal{N})$.

Theorem 11. *There is a nonstandard model for $Th(\mathcal{N})$, i.e. there exists a model $\mathcal{M}$ such that*

1. $\mathcal{M} \models Th(\mathcal{N})$ *and*
2. $\mathcal{M} \not\cong \mathcal{N}$.

Proof. Let L be the vocabulary of standard model of natural number, $L' = L \cup \{c\}$, where c is a new symbol. Consider the theory

$$\Delta = Th(\mathcal{N}) \cup \{c \neq n \mid n \in \mathbb{N}\}.$$

In other words, c is different from any natural number. Suppose Δ' is a finite subset of Δ, if it does not contains c, it must be satisfiable; otherwise we can interpret c as any integer that is larger than the any integers used in $Th(\mathcal{N})$, to obtain a model satisfying Δ'. By the compactness theorem, there exists a model $\mathcal{M}$ for Δ.

Clearly $\mathcal{M}$ is also a model for $Th(\mathcal{N})$. Moreover, since c is different from any natural numbers, $\mathcal{M}$ is not isomorphic to $\mathcal{N}$. In other words, there exists a nonstandard model for $Th(\mathcal{N})$. □

Note that this proof is not constructive, since the proof of compactness theorem is not constructive. The construction of a nonstandard model for natural number is much more complicated.

6.5 Löwenheim-Skolem Theorem

Another important result about model is called the Löwenheim-Skolem Theorem.

Theorem 12 (Löwenheim-Skolem Theorem)**.** *Suppose Δ is a set of sentences. If Δ has a model, it has a countable model. Moreover, it it has an infinite model, it has a countably infinite model.*

Proof. Suppose Δ has a model, we know that it must be consistent. Then the proof of Theorem 9 explicitly gives a countable model of it.

Let Δ' be

$$\Delta \cup \{\exists_{\geq n} x\ x = x \mid n = 1, 2, \ldots\},$$

where $\exists_{\geq n} x \varphi(x)$ means that there are at least n distinct elements c such that $\varphi(x \mapsto c)$ is satisfied. Clearly $\exists_{\geq n}$ can be easily defined for any fixed integer n.

Suppose Δ has an infinite model, Δ' also has an infinite model. By Theorem 9 we can give a countable model $\mathcal{M}$ of Δ', which is also a model of Δ. Since $\mathcal{M} \models \exists_{\geq n} x\ x = x$ for any integer n, it must be a countably infinite model. □

An application of Theorem 12 is to provide a criteria of the completness of a theory. Such method is called the Vaught's Test.

Theorem 13. *If all models of T is infinite and it is countably categorical, then T is complete.*

Proof. Suppose $\mathcal{M}$ and $\mathcal{M}'$ are two models of T. Let $T_1 = Th(\mathcal{M})$, $T_2 = Th(\mathcal{M}')$, by Theorem 12, there must be an infinite countable models $\mathcal{M}^*$ and $\mathcal{M}'^*$ for T_1 and

T_2, respectively. Clearly they are also models for T. Since T is countably categorical, $\mathcal{M}^* \cong \mathcal{M}'^*$. Since $Th(\mathcal{M})$ and $Th(\mathcal{M}')$ are complete, it's obvious that

$$\mathcal{M} \equiv \mathcal{M}^*,$$
$$\mathcal{M}' \equiv \mathcal{M}'^*,$$

so that $\mathcal{M} \equiv \mathcal{M}'$. As a result, all the models of T are elementarily equivalent, thus T must be complete. $\square$

Vaught's Test can be used to study the completeness of some concrete theories. For example, let T be the theory of equivalence relation, and

$$T' = T \cup \{\exists_{\geq n} x\ x = x\} \cup \{\forall x \exists_{=2} y\ R(x, y)\},$$

Then T' represents the theory in which

1. R forms an equivalent relation;
2. there are infinite elements;
3. each block in the partition induced by R contains two elements.

Note that $\exists_{=2} y \varphi(y)$ can be easily defined in first-order logic.

Clearly T' has only infinite models, and countable models of T' are isomorphic, since the structure of partition induced by the equivalent relation is fixed. By the Vaught's Test, we can deduce that T' is complete.

7 Incompleteness

In this section we will show a proof outline of the famous Gödel's Incompleteness Theorem. During the proof, we will use a property about computable function, so it is expected that the readers are familiar with the Turing machines (or λ calculus) and its computing ability.

7.1 Peano Arithmetic

We have already see that $Th(\mathcal{N})$ is not practical to study the property of natural numbers, so it's reasonable to construct a formal system for it. Peano Arithmetic is the most famous approach of the axiomatization of natural numbers.

Let $L_{PA} = \{0, ', +, \cdot\}$, we can define the following axioms for the operators.

1. $\forall x \neg(x' = 0)$.
2. $\forall x \forall y (x' = y' \to x = y)$.
3. $\forall x\ x + 0 = x$.
4. $\forall x \forall y (x + y' = (x + y)')$.
5. $\forall x\ x \cdot 0 = 0$.
6. $\forall x \forall y (x \cdot y' = x \cdot y + x)$.

Moreover, an additional axiom scheme, the principle of induction, is added in the system.

- $\varphi(0) \to \forall x(\varphi(x) \to \varphi(x')) \to \forall x \varphi(x)$, for any formula φ with only one free variable.

It can be expressed in natural language as: *if a property holds for* 0*, and* x *has the property implies* $x+1$ *has the property, then any natural number has the property.* We have been quite familiar with such principle, since we have already used a generalized version of it (structure induction).

With the external axiom scheme of mathematical induction, we can create a new proof system PA by adding the axioms into the Hilbert-style system. If a sentence φ is provable in such system, it is called a theorem in peano arithmetic, denoted as PA $\vdash \varphi$.

Note that such method is called axiomatization (in first-order logic) of a mathematical structure. The axioms added into the system are called nonlogical axioms.

7.2 Expressibility of PA

A relation on natural number $R(x_1, \ldots, x_n)$ is expressible, if and only if there exists a formula $\varphi(x_1, \ldots, x_n)$ with n free variables in Peano arithmetic, such that

$$\mathrm{PA} \vdash \varphi(x_1 \mapsto 0^{(m_1)}, \ldots, x_n \mapsto 0^{(m_n)})$$

if and only

$$R(m_1, m_2, \ldots, m_n),$$

where $0^{(m)}$ denotes $0^{''\cdots'}$, the representation of m in PA.

Similarly, we can define the expressibility of set by considering relation $R(x)$, and expressibility of elements by considering expressible singleton set.

For a relation R over natural numbers, the characteristic function of R, denoted as $c_R(n_1, \ldots, n_m)$ is defined as

$$\begin{aligned} c_R(n_1, \ldots, n_m) &= 1, \ \text{ if } R(n_1, \ldots, n_m) \\ c_R(n_1, \ldots, n_m) &= 0, \ \text{ otherwise.} \end{aligned}$$

We will show that the expressibility of a relation depends on the computability of the characteristic function of it.

A function f is called primitive recursive, if it is a basic function, or it is generated by applying two rules for finite number of times. There are three (classes of) basic functions.

1. Zero function. $f(n) = 0$, for all $n \in \mathbb{N}$.
2. Successor function. $f(n) = n + 1$, for all $n \in \mathbb{N}$.
3. Projection function. $p_i^k(n_1, \ldots, n_k) = n_i$.

And there are two generation rules.

- Composition. Suppose $g(n_1, \ldots, n_j)$ is a function and $h_i(n_1, \ldots, n_k)$ are functions for all $1 \le i \le j$, then we can construct

$$\begin{aligned} &f(n_1, \ldots, n_k) \\ =&g(h_1(n_1, \ldots, n_k), \ldots, h_j(n_1, \ldots, n_k)). \end{aligned}$$

- Recursion. Suppose $g(n_1, \ldots, n_k)$ and $h(n_1, \ldots, n_{k+2})$ are functions, we can construct a function f, such that

$$\begin{aligned} & f(n_1, \ldots, n_k, 0) \\ =& g(n_1, \ldots, n_k) \\ & f(n_1, \ldots, n_k, m+1) \\ =& h(n_1, \ldots, n_k, m, f(n_1, \ldots, n_k, m)). \end{aligned}$$

Note that it's easy to see that a primitive recursive function is well-defined for all natural numbers. Moreover, a primitive recursive function must be computable, i.e. there is a Turing machine to compute such function. But the converse is not correct.

Theorem 14. *There exists a computable function that is not primitive recursive.*

Proof. Clearly we can encode all the primitive recursive functions into natural numbers, denoted as $\{p_0, p_1, \ldots\}$. Now consider the function

$$f(n) = p_n(n) + 1,$$

clearly such function is computable, since Turing machine is strong enough to simulate the primitive recursive function encoded by natural number. Moreover, such function is different from p_i for all i, so that f is not primitive recursive. □

This proof technique is called the diagonalization method.

Note that primitive recursive function is sufficiently strong to express the following functions and concepts in computation.

1. Factorial function. $x \mapsto x!$.
2. The predecessor function. $x' \mapsto x$, $0 \mapsto 0$.
3. The sign function. $x' \mapsto 1$, $0 \mapsto 0$.
4. Boolean operations and bounded quantifications, for example, $\varphi \vee \psi$, $\exists y_{y \leq z} \varphi(y)$. Note that $\exists y_{y<z} \varphi(y)$ means that there exists a natural number n smaller than z such that $\varphi(y \mapsto n)$ is correct.
5. List and its operation. A list is denoted as $[x, y, \ldots, z]$, thus concatenate, head function $[x, y, \ldots, z] \mapsto x$, tail function $[x, y, \ldots, z] \mapsto z$, length function and some other basic operations to lists that can be defined recursively.

We can further extend the computation ability of primitive recursive function by adding regular minimization rule.

- Suppose $f(x, y_1, \ldots, y_m)$ is a function such that for all $n_1, \ldots, n_m$, there exists some k such that $f(k, n_1, \ldots, n_m) = 0$. Then we can construct a new function denoted as $\mu_x f(x, y_1, \ldots, y_m)$, which finds the smallest k such that $f(k, n_1, \ldots, n_m) = 0$. In other words, it performs an unbounded search of valid x.

A function generated from basic functions and composition, recursion and regular minimization rules is called recursive.

Theorem 15. *A function is recursive if and only if it is (Turing) computable.*

Proof. By constructing interpreter of Turing machine with recursive function and vice versa. Details omitted. □

Now we wound like to introduce the theorem which connects expressibility and computability.

Theorem 16. *Suppose $f(x_1, \dots, x_n)$ is a recursive function, then the relation $f(x_1, \dots, x_n) = y$ is expressible in Peano Arithmetic. In particularly, if the characteristic function of a relation is recursive, then the relation is expressible.*

We will give a proof sketch of the theorem, and refer our readers to the textbooks of mathematical logic, i.e. OpenLogic (see 8).

The second part of the proof is the direct consequence of the first part. Suppose $c_R(n_1, \dots, n_m)$ is recursive, by the first part of the theorem, we know that $c_R(n_1, \dots, n_m) = 1$ is expressible, clearly this is just the representation of R.

The proof of the first part is done by structural induction on the generation of recursive function. Before doing that, we may use the following lemma to remove the recursion rule, in order to simply the proof.

Lemma 14 (Beta-Function Lemma). *For any sequence of natural number $a_0, \dots, a_n$, there exists a function $\beta(D, u)$ constructed by composition and regular minimization from basic functions, such that for $1 \leq i \leq n$, $\beta(D, i) = a_i$.*

Proof. This construction based on the Chinese Remainder Theorem in elementary number theory. Suppose the sequence is $a_0, a_1, \dots, a_n$, let $m = \max\{n, a_0, \dots, a_n\} + 1$, consider the sequence

$$\begin{aligned} x_0 &= 1 + m!, \\ x_1 &= 1 + 2m!, \\ &\dots \\ x_n &= 1 + (n+1)m!. \end{aligned}$$

Firstly we will show that x_i and x_j are relatively prime for all $i \neq j$. Without loss of general suppose $i < j$, by the Euclidean algorithm, we know that

$$\begin{aligned} \gcd(x_i, x_j) &= \gcd(x_j, x_j - x_i) \\ &= \gcd(1 + (j+1)m!, (j-i)m!). \end{aligned}$$

For any prime divisor d of $1 + (j+1)m!$, it must not be a divisor of $m!$, so that it can neither divide $(j - i)$, thus it can not divide $(j - i)m!$. As a result, we know that $\gcd(x_i, x_j) = 1$, which means that $\{x_i\}$ are pairwisely relatively prime.

Furthermore, we know that $a_i < x_i$ for all $0 \leq i \leq n$, thus by the Chinese Remainder Theorem, there exists a natural number $0 \leq d < x_0 x_1 \dots x_n$ such that

$$\begin{aligned} a_0 &= d \bmod x_0, \\ a_1 &= d \bmod x_1, \\ &\dots \\ a_n &= d \bmod x_n. \end{aligned}$$

We will use the pair (d, m) encoded by natural number as the first parameter of β function. It can be easily checked that if (d, m) is given, we can write a function β with only composition and minimization rules, such that

$$\beta((d, m), i) = d \bmod x_i,$$

which is what to prove. $\square$

Then we can simulate recursion with composition and regular minimization, by *guessing* the sequence encoded by natural number. More preciously, Suppose $g(n_1, \ldots, n_k)$ and $h(n_1, \ldots, n_{k+2})$ are two functions, the function generated by recursion rule, denoted as $f(n_1, \ldots, n_k, m)$ is equivalent to the following one.

Let $d_0, d_1, \ldots, d_m$ be the sequence defined as

$$\begin{aligned} d_0 &= g(n_1, \ldots, n_k) \\ d_{i+1} &= h(n_1, \ldots, n_k, i, d_i), \end{aligned}$$

which means that it is the sequence of computation during the recursion. By the Beta Function Lemma, we know that there exists β and the encoding D for the sequence. Let $f'(n_1, \ldots, n_k, m)$ is defined as

$$\begin{aligned} &\mu_D(\beta(D, 0) = g(n_1, \ldots, n_k) \wedge \\ &\quad \forall 0 \le i < n, \\ &\quad \beta(D, i+1) = h(n_1, \ldots, n_k, i, \beta(D, i))). \end{aligned}$$

Intuitively speaking, it guesses the recursion process and then verify it.

Note that the second part of the verification process

$$\begin{aligned} &\forall 0 \le i < n, \\ &\beta(D, i+1) = h(n_1, \ldots, n_k, i, \beta(D, i)) \end{aligned}$$

is definable with some tricks using only regular minimization and composition. For example, we can guess a sequence of length $m+2$, and verify that the first i such that

$$\beta(D, i+1) \neq h(n_1, \ldots, n_k, i, \beta(D, i))$$

is $m+1$, the unused term which can be arbitrarily chosen. This construction allows us to consider only composition and regular minimization in proving Theorem 16.

Now we will finish the proof of the theorem. Suppose $\varphi(f, x, y)$ denotes the representation of $f(x) = y$. Clearly, for any basic functions f, $f(x) = y$ must be expressible (see Table 5).

Function	Representation
Zero	$y = 0$
Successor	$y = x'$
Projection	$y = n_i$

Table 5: Representation of basic functions.

For composition rule, we can use existential quantifier to "guess" the result of the functions $h_1, h_2, \ldots, h_j$, to verify the result of the composed function.

For regular minimization rule, we can simply interpret its meaning in first-order logic, as

$$f(y, x_1, \ldots, x_n) = 0$$
$$\forall i < y, f(i, x_1, \ldots, x_n) \neq 0.$$

By induction hypothesis, we know that both $f(y, x_1, \ldots, x_n) = 0$ and $f(i, x_1, \ldots, x_n) \neq 0$ are expressible.

By carefully checking, we can show that such construction is valid, thus any recursive function (also computable functions) can be expressed in PA.

7.3 The Fixpoint Lemma

Now we would like to introduce one of the most important lemmas during proving the Incompleteness Theorem and related results.

Lemma 15 (The Fixpoint Lemma). *Suppose $[*]$ is some computable encoding method from the formulas in PA to natural numbers, $\varphi(x)$ is some formula with the only free variable x, then there exists a sentence ψ such that* $\mathrm{PA} \vdash \psi$ *if and only if* $\mathrm{PA} \vdash \varphi(x \mapsto [\psi])$.

Proof. Let $\mathrm{app}([\varphi(x)], [\psi])$ be a function such that it generate $[\varphi(x \mapsto [\psi])]$ where ψ is a formula. Clearly it is computable, thus $\mathrm{app}(x, y) = z$ is expressible.

Let $\psi'(x)$ be the formula with only one free variable x, defined as

$$\psi'([\rho]) = \exists z(\mathrm{app}([\rho], [\rho]) = z \wedge \varphi(x \mapsto z)).$$

Since ψ' is a formula, it has some encoding $[\psi']$ which is a concrete natural number, let $\psi = \psi'(x \mapsto [\psi'])$. By the definition, we know that

$$\psi = \exists z(\mathrm{app}([\psi'], [\psi']) = z \wedge \varphi(x \mapsto z)).$$

If $\mathrm{PA} \vdash \psi$, then we know that

$$\mathrm{PA} \vdash \varphi(x \mapsto \mathrm{app}([\psi'], [\psi'])),$$

which is exactly $\varphi(x \mapsto [\psi])$. Conversely if $\mathrm{PA} \vdash \varphi(x \mapsto [\psi])$, we can also know that

$$\mathrm{PA} \vdash \psi,$$

which means that ψ is the fixpoint of φ. □

Note that the proof of the fixpoint lemma is similar to the proof of the recursion theorem in theory of computation.

A direct consequence of the fixpoint lemma is Tarski's theorem of the undefinability of truth.

Theorem 17 (Tarski's Theorem of Undefinability of Truth)**.** *There is no definition of truth in PA. More preciously, let the relation* $\mathrm{True}([\varphi])$ *hold if and only if* $\mathrm{PA} \vdash \varphi$, *then* R *is not expressible.*

Proof. If it is expressible, i.e. there exists a formula $\varphi(x)$ with the only free variable x to express the truth relation. By the fixpoint lemma, we know that there exists a sentence ψ such that $\mathrm{PA} \vdash \psi$ if and only if $\mathrm{PA} \vdash \neg\varphi(x \mapsto \psi)$, which leads to a contradiction. □

7.4 Incompleteness Theorems

Theorem 18 (Gödel's Incompleteness Theorems)**.** *Assume* PA *is consistent, then*

1. PA *is not complete;*
2. PA *can not prove its own consistency, i.e. it can not prove that* $\mathrm{PA} \nvdash \bot$.

Proof. Assume that PA is consistent. For the first part, clearly we can also encode the proofs in system H with natural numbers, also denoted as $[*]$. Let $\mathrm{Prf}([\varphi], [pf])$ be the relation such that pf is a proof for φ. It's easy to see that the characteristic function of the relation can be computed by Turing machine, so that Prf is expressible in PA. By the fixpoint lemma, we know that there exists ψ such that $\mathrm{PA} \vdash \psi$ if and only if

$$\mathrm{PA} \vdash \forall x \neg \mathrm{Prf}([\psi], x).$$

If $\mathrm{PA} \vdash \psi$, by the definition we know that there is no proof for ψ; otherwise if $\mathrm{PA} \vdash \neg\psi$, we know that there is some proof for ψ. As a result, both ψ and $\neg\psi$ are not provable in PA, thus PA is incomplete.

For the second part, we may restate the argument as $\mathrm{PA} \nvdash \neg\mathrm{Prf}(\bot)$. Let γ be the sentence for the first part of the proof, and φ be the sentence $\neg\mathrm{Prf}(\bot)$, we will show that $\mathrm{PA} \vdash \varphi \to \gamma$. Since γ is not provable in PA, we know that φ is also not provable. The high level idea of the proof in system H is given in the following table.

Proof	Reason
$\gamma \to \neg\mathrm{Prf}(\gamma)$	First part
$\gamma \to \mathrm{Prf}(\gamma) \to \bot$	Definition
$\mathrm{Prf}([\gamma \to \mathrm{Prf}(\gamma) \to \bot])$	P1
$\mathrm{Prf}([\gamma]) \to \mathrm{Prf}([\mathrm{Prf}(\gamma)]) \to \neg\varphi$	P2
$\mathrm{Prf}([\gamma]) \to \mathrm{Prf}([\mathrm{Prf}(\gamma)])$	P3
$\mathrm{Prf}([\gamma]) \to \neg\varphi$	Logic
$\varphi \to \mathrm{Prf}([\gamma])$	Logic

We use three properties of Prf, denoted as P1, P2 and P3, which means that

1. $\mathrm{PA}, \gamma \vdash \mathrm{Prf}([\gamma])$, by the definition of expressibility;
2. $\mathrm{PA}, \mathrm{Prf}(\varphi \to \psi) \vdash \mathrm{Prf}(\varphi) \to \mathrm{Prf}(\psi)$, by deduction theorem and the definition of Prf;
3. $\mathrm{PA} \vdash \gamma \to \mathrm{Prf}([\gamma])$, by P1 and deduction theorem.

□

Note that the consistency of PA is proved by Gentzen in 1935, four years later then Gödel established his incompleteness theorems. So now we can remove the assumption that PA is consistent and conclude that PA can not be complete.

We may also informally argue that extending the Peano Arithmetic (i.e. adding finite number of axioms) will not help to solve the problem of incompleteness as long as it remains consistent, which is proved by Rosser.

8 Reference

If you find any mathematical incorrectness, typos or lack of precision in description, it is welcomed to report it to me by email.

SlideFOL Dag Westerståhl, *Lecture Slides: Foundation of Logic* (2020 Spring), at Tsinghua University (30690552-0).

SlideTOC Duan Ran, *Lecture Slides: Theory of Computation* (2020 Spring), at Tsinghua University (30470134-0).

OpenLogic *The Open Logic Text*, `https://openlogicproject.org/`.

NoteCH *Lecture Notes on Curry-Howard Isomorphism*, Morten Heine B. Sørensen, Pawel Urzyczyn.