A Mathematical Introduction To Logic

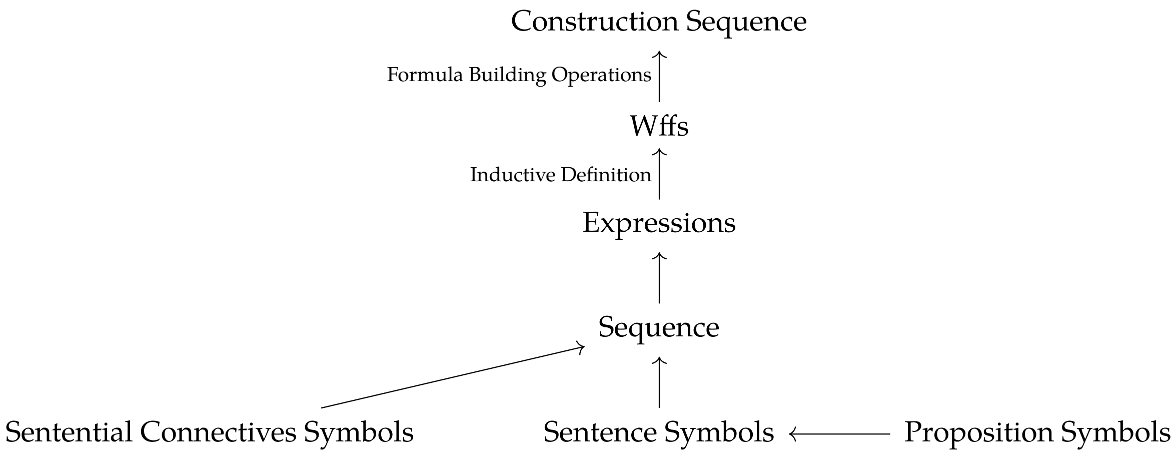
Herbert Enderton Notes taken by Dendy Li

November 12, 2024

Contents

1 Sentential Logic

1.1 Informal remarks on the basic objects



Definition 1.1 (Sentential Connective Symbols)**.** Sentential connective symbols is the symbols which have fixed translations.

eg. We have five sentential connective symbols in sentential logic namely

- $\neg$
- Conjunction $\wedge$
- Disjunction $\vee$
- Implies $\rightarrow$
- Iff $\leftrightarrow$

Definition 1.2 (Sentence Symbols)**.** Sentence symbols are the symbols which doesn't fixed the translation.

Remark: We also have proposition symbols for some case we want the translation of them are propositions

Definition 1.3 (Expressions)**.** Expressions just finite sequence of symbols(both sentential connective symbols and sentence symbols).

Remark: In order to avoid the Russuell's Paradox we assume that no symbol is a finite sequence of symbols

Definition 1.4 (Well Form Formula)**.** Well form formulas are just Grammarly correct expressions. By Grammarly correct we means the inductive definition of the well form formula.

Proposition 1.5 (Inductive definition of wffs)**.** *All the wffs can only be obtained by the following rules*

- *Every sentence symbol is a wff*
- *If α is a formula then $\neg\alpha$ is a wff*
- *If α, β are wffs then $\alpha \circ \beta$ is a wff*

Here $\circ$ represents the other 4 sentential connectives we didn't mention above.

We will then define the formula building operations to show how wff is built up.

Definition 1.6 (Formula building operations)**.** For wff α, β, formula building operation is defined as follows:

- $\mathrm{E}_{\neg}(\alpha) = (\neg\alpha)$
- $\mathrm{E}_{\circ}(\alpha, \beta) = (\alpha \circ \beta)$

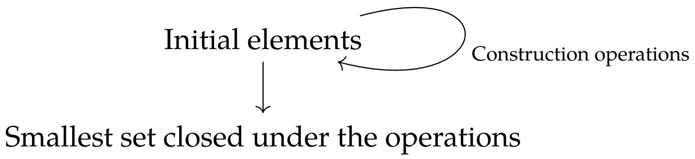
Here $\circ$ is one of the other 4 sentential connectives.

Definition 1.7 (Construction Sequence of a wff)**.** The construction sequence of a wff α is a sequence $< a_1, \ldots, \alpha >$, where every a_i is obtained by element in the front of a_i under the formula building operations.

1.2 Induction

Induction is a special type of construction, especially when we face the problem that the inductive defintion brings to us.

Here we have "Universal" progress on meta-strategy of how induction works: **Goal: Starting from a set U, construct a subset of U satisfies certain properties.**



Remark: Every members in the "Gerating Set" can be build up from the initial elements by apply the construction operations. This also arise the interests to talk about the definition of free generation, namely whether the construction sequence is unique or not.

eg. We can have an easy understandable case where the initial elements are just the sentence symbols, the operations are formula building operations. In this case the set we generate is just the set of well-formed-formulas.

Next we want to formally discuss this "Smallest Set". For convinient we use the following conventions:

- Initial set $B \subseteq U$
- Class of operations $\mathcal{F} := \{f, g\}$, where $f : U \times U \to U, g : U \to U$. Notice that in a general case the class of operations should be infinite.

With the notation above we use a little more formal language to represent the process:
WLOG suppose B contains points a, b, then C will contain the elements in the form of:

$$b, f(b, b), g(b), f(g(a)), ...$$

Now we formally consider this problem since this construction process can be described by two ways:

- From bottom to the top(C_*): We define C_* to be the set generated by B with $\mathcal{F}$. Here by generated we mean the generating process above.
- From top to the bottom(C^*): We define C^* to be the intersection of all subsets of B closed under the operations in $\mathcal{F}$
- We show these two sets are actually the same.

Here one might meet two definitions which is important:

Definition 1.8 (Closed under operations)**.** Let S be a set closed under operations, then for any elements of S, their interactive(combinition for all) image under those operations are contained in S.

Definition 1.9 (Inductive Set)**.** For initial set B, we say a set S is inductive if:

- $B \subseteq S$
- S is closed under the given operation class.

By all the above conventions one can reach the following theorem:

Theorem 1.10. $C^* = C_*$.

Now we can formally describe the induction principle:

Theorem 1.11 (Induction Principle)**.** *Let C is the set generated from B by the operations in $\mathcal{F}$. For any set S lies between B, C, with the assumption that it closed under operations in $\mathcal{F}$, then $S = C$.*

1.3 Recursion

1.3.1 Freely Generated Sets

Because one needs to define functions recursively over an inductive closure, the freely generated property allows us to unique reprsent the elements in the generated set by a tree.

eg. In the process of evaluating the expressions we first define a function which enables us to evaluate the atoms, the we recursively define the extension of the function to evaluate all expressions. By free generation we can make sure that the tree for the evaluation process for each expression is unique.

Definition 1.12 (Freely Generated Sets)**.**

A problem we care about: How to recursively define a function? Basically we need 3 functions

- g:= "Push forward function"
- f:= "Form of base case function"
- h:= "Recursive defined function"

with few rules on how these functions are interacting with each other.

1.4 Compactness and Effectiveness

Convention: Let Σ denote a set of wffs.

Theorem 1.13 (Compactness Theorem)**.** *Σ is satisfiable iff every finite subset of Σ is satisfiable.*

Remark: This theorem is called compactness theorem since it is kind of looks like the compactness defined in analysis.

Definition 1.14 (Finitely satisfiable(Temporarily definiton))**.** Σ is finitely satisfiable iff every finite subset is satisfiable.

Proof Sketch.

- we first use the given finitely satisfiable set to extend it to the maximal.
- Then we construct a truth assignment out of the maximal set we construct above to satisfies Σ.

□

2 First-Order Logic

Propositional Logic is a basic model for us to formally express our informal mathematics. However, Propositional Logic can't describe the quatifiers, by adding the quatifiers and inherits some ideas of Propositional Logic, we have the first-order Logic, which allows us to formally talk about more mathematical structures.

Some important features from the Chapter 1: Notice that $\{\neg, \rightarrow\}$ is complete so we want out sentential connectives only include these two symbols. Also, notice that $\exists = \neg\forall\neg$, we only want to include the universal quantifier.

Getting familiar with some basic ideas of first-order logic:

- Socretes is a man, here "is a man" is a predicate.
- Metamathematics refers to the procedure of stepping back and examining what the mathematician is doing.

2.1 First-order language

2.1.1 Expressions, Terms, Atomic Formulas and Well-formed formulas

Assumption: we are given infinitely many distinct symbols as follows:

- Logical Symbols
  - Parentheses
  - Sentential connective symbolsa
  - Variables(countably many)
  - Equality symbol(optional)
- Parameters
  - Quantifier symbol
  - Predicate
  - Function symbol
  - Constant symbol(0-place function symbol)

Remark: All symbols appear here are all distinct, Logical Symbols and Parameters are determinated we say the language is been determinated.

Proposition 2.1 (Distinctions between languages)**.** *Languages are only distinct in parameters and equalities.*

Next we look at the language of set theory and the language of the elementary number theory to see why large mathematics can be embedded into set theory.

Language of set theory	Language of elementary number theory
Predicates : yes	Predicates : yes
Functions : no	Functions : yes
Constants : yes	Constants : yes
Equality : yes	Equality : yes

I omit the concrete type of the parameters here, but the reader can have a feeling that the language of set theory can somehow embed in the language of elementary number theory. Just like Group can be embed in a ring.

We have the following informal expression about the statement above: Large mathematcis can be embedded into set theory because they can be expressed by set theory and mathematics theorem follow logically from the axioms of set theory.

Here is a unintuitive phenomenon of first-order logic: eg.

$$\begin{aligned} &\exists v_1(Av_1 \wedge Bv_1) && \text{There is an object } v_1, \text{ such that } v_1 \text{ is an apple and } v_1 \text{ is a bad apple.} \\ &\exists v_1(Av_1 \to Bv_1) && \text{There is an object } v_1, \text{ such that if } v_1 \text{ is an apple, then } v_1 \text{ is a bad apple.} \end{aligned}$$

The second formula is more asserting than the first formula, since the first formula will fail in the universe where there is no apple exists while the second formula still remaning true. (Because the second formula has the assumption if the object is an apple.)

Few words which need a little declaration:

- nonsensical: no meaning
- assertion: confindent statement
- intend to: want to or plan to

Definition 2.2 (Expression)**.** An expression is a finite sequence of symbols.

Not all the expressions have meaning, we might stick to some of the meaningful sentences-Terms, Atomic formulas, Well-formed formulas.

We have the following correspondence:

$$\text{Nouns and Pronouns} \leftrightarrow \text{Terms} \tag{1}$$
$$\text{Sentencial symbols} \leftrightarrow \text{Atomic formulas} \tag{2}$$

Comparing with the natural languages, we want the term to be the name of the objects and well-formed formulas to be the assertions.

Construction of meaningful expressions:

- Construction of terms
  - For every function f, there is an induced formula building operation $\mathcal{F}_f$ where compound n variables into one term(for n-placed function symbol f).
  - **Definition of the term:** Every term is of the form $\mathcal{F}_f$(sequence of constant symbols or variables).
  - One special case is that, there are no function symbol in the language, so the terms are just constant and variables.
  - Syntax of the term:
    - ∗ Polish notation
    - ∗ No parentheses(Unique readability theorem ensure the well-define)
- Construction of Atomic formulas
  - Atomic formulas aren't defined by induction, it's just a sequence starting with a n-placed predicate symbol follows by n terms.
- Construction of Well-formed formulas

- Well-formed formulas are expressions which are built from the atomic formulas by the defined formula building operations on the expressions(For implication, negation, universal quantifiers).

  Remark: Atomic Formulas are wffs.

2.1.2 Free Variables

In natural languages, the following examples express the idea of free variables:

$$\forall v_2 \in v_2 v_1 \text{ Every set is a number of ( )}$$
$$(\neg\forall v_1(\neg\forall v_2 \in v_2 v_1)) \text{ There is a set } v_1, v_1 \text{doesn't contain every } v_2 \text{ as a subset.}$$

Remark: Free variable just likes "Blank" in our everyday language, next we want to formally define what is a free variable since we don't want our definition relies on the translation in Natural language.

In order to directly define what is a free variable, we first define what does a variable occurs free in a given formula means, then a variable occus free is a free variable.

Definition 2.3 (Occur free)**.** Let ϕ, φ be wffs, α is atomic formula:

- For atomic formula α, x occurs free in α;
- For formula $\neg\varphi$, x occurs free in $\neg\varphi$ iff x occurs free in φ;
- For formula $\varphi \to \phi$, x occurs free in $\varphi \to \phi$ iff x occurs free in φ and ϕ;
- x occuts free in $\forall y\varphi$ iff $x \neq y$ and x occuts free in φ.

  We say x is a free variable in formula ϕ if x occurs free in ϕ.

The definition above is valid because we can first define the occur free on atomic formulas and basic construction of atomic formulas, then by recursion theorem we can extend the definition on atomic formulas to wffs.

Definition 2.4 (Sentence)**.** A wff without any free variable is a sentence.

This definition is intuitively valid, since when we talk to each other we normally don't leave blank while talking.

2.1.3 Conventions for readability of First order language

In the definition of First order language, we don't use $\wedge, \vee, \neq, ...$, but this will make the formulas hard for us to read. We have the following convention:

- Negation and Universal quantifiers control the smallest formula next to it;
- Repeat expressions group to right.

2.2 Truth and Models

$$\text{Sentencial Logic} \Leftarrow \text{Truth assignment(Tell truth by the value)}$$
$$\text{First order Logic} \Leftarrow \text{Structure(Tell truth by checking the dictionary)}$$

Structure is a dictionary for us to check the interpretations of the formulas.

Definition 2.5 (Structure)**.** Structure is a function which can be described as follows:

$$\begin{aligned}\forall &\mapsto |\mathfrak{A}|\\ \text{n-ary Predicate symbols} &\mapsto \text{Relations on } |\mathfrak{A}|^n\\ \text{n-ary Function symbols} &\mapsto \text{Operations on}|\mathfrak{A}|^n\\ \text{Constant symbols } c &\mapsto c^{\mathfrak{A}} \in |\mathfrak{A}|.\end{aligned}$$

Here we recall a little bit about the definition of relation and operations to have a clear understanding of what does the structure $\mathfrak{A}$ looks like.

Definition 2.6 (n-ary relation and n-ary operation)**.** n-ary opertaion on A is a function:

$$F : A^n \to A,$$

where any subset of A^n is a n-ary relation on A.

One might want to ask: What's the difference between a function and a relation, we have the follow informal definition:

Definition 2.7 (Function)**.** F is called a function if:

- F is a relation;
- No two distinct pairs in F have the same first coordinate.

So a n-ary function is somehow a $n+1$-ary relation with additional property(first coordinate unrepeated).

Informally a structure $\mathfrak{A}$ can be viewed as a set with different forms of elements. There are tuples to represent the relation, functions, there are "Points" to represent the universe where the constant symbols lies in.

Here we might remind myself a little bit on what's the relationship between the structure and the language. The structure plays the row of the truth assignment where the language is for how to write the grammarly correct sentences.

$$\begin{tikzcd}
\text{Informal Mathematics} \arrow[r, "embedded"] \arrow[dr, "Judge"'] & \text{First order language of it}(L) \arrow[r, "Produce"] & \text{Wffs} \arrow[d, "Input"] \\
& \text{Informal statements} & \text{Structure of } L \arrow[l, "Interpretate"]
\end{tikzcd}$$

eg. In the first order language for set theory, take $\mathfrak{A} = (|\mathfrak{A}|, \in^{\mathfrak{A}})$ where $|\mathfrak{A}| := \mathbb{N}$ and $\in^{\mathfrak{A}} := \{< m, n > |m < n, m, n \in \mathbb{N}\}$. Given a formula in first order language of set theory, we can translate it into the informal statement and use our knowledge about set theory to check whether the sentence is true or false. **If this sentence φ is true, we say the structure $\mathfrak{A}$ is a model of $varphi$.**

One little remark is the first order language of set theory doesn't mean the set theory we use in the informal mathematics, a first order language of set theory is any language equipped with a universe which has a contant symbol for empty set and a 2-ary predicate for inclusion.

One problem of the translation above is that we can't handle the truth of the self-reference sentences, like "This sentence is false.". This is the motivation for a formal definition of what do we mean by a sentence is true under the structure $\mathfrak{A}$.

2.2.1 Define Truth by structure

Like in sentential logic, we define the "truth" with respect to "Truth Assignment". Here "Truth Assignment" we simply mean a "Translation in the Model".

2.3 Deductive Calculus

2.3.1 Formal Deductions

Basic ideas of deduction:

$$\begin{tikzcd} \text{Axioms } + \text{Assertions} \arrow[r] \arrow[d] & \text{Theorems} \\ \Lambda \cup \Gamma + \text{Deduction} \arrow[r] & \text{Theorems} \end{tikzcd}$$

In the second line, a theorem is just a formula obtain from $\Lambda \cup \Gamma$ by the rule of inference, which we call deduction here.

Definition 2.8 (Deduction)**.** Let Λ be a set of infinitely many fomulars, we usually call it axioms. A deduction is a finite sequence of formulas s.t. every element in the sequence can be interfere by earlier ones or this element lies in the axioms or theorems.

Theorem 2.9 (Generalization Theorem)**.** *If $\Gamma \vdash \varphi$ and $\forall \alpha \in \Gamma,\ x \notin Free(\alpha)$, then we can add universal quantifier before formula φ, namely:*

$$\Gamma \vdash \forall\ x\varphi.$$

Definition 2.10 (Six groups of Logical axioms)**.** The following are six groups of Logical axioms:

- Generalization of tautologies;
- Generalization of $\forall x\alpha \to \alpha^x_t$, t is substitutable for x in α;
- Generalization of $\forall x(\alpha \to \beta) \to (\forall x\alpha \to \forall x\beta)$;
- Generalization of $\alpha \to \forall x\alpha$;
- Generalization of $x = x$;
- Generalization of $x = y \to (\alpha \to \alpha'$, where α' obtained from α by replacing x in 0 or more occurrences by y.

Next we give some detailed consideration of the logical axioms above.

2.4 Completeness Theorem

The completeness theorem can give us aan answer about the equivalence of $\vDash$ and $\vdash$. Then equivalence them will be given in two different theorems namely soundness theorem and completeness theorem.

2.4.1 Soundness theorem

Theorem 2.11 (Soundness theorem)**.** *If $\Gamma \vdash \varphi$, then $\Gamma \vDash \varphi$.*

2.4.2 Completeness theorem

Theorem 2.12 (Completeness theorem). *If* $\Gamma \vDash \varphi$*, then* $\Gamma \vdash \varphi$*.*

We will not follow Gödel's proof of the completeness theorem. We will use the Henkin Construction.

Here's a sketch of how the proof goes:

$$\begin{tikzcd}
\Gamma \arrow[r] \arrow[d] \arrow[dr] & \mathscr{L} \arrow[d] \\
\Gamma \cup \Theta \arrow[r] \arrow[d] & \mathscr{L}' \arrow[d] \\
\Delta \arrow[r] \arrow[ur] \arrow[dr] & A \arrow[d] \\
 & A/E
\end{tikzcd}$$

We fixed a countable language $\mathscr{L}$ and a set of consistent formulas Γ.

- Step-1: Extend the Language $\mathscr{L}$ by adding countably infinite constant symbols, the set Γ still consistant in $\mathscr{L}'$;
- Step-2: Extend the set of consistant formulas by adding formulas of the form,

  $$\neg\forall x\varphi \to \neg\varphi^x_c,$$

  which gives a enumeration of the formulas by the constant symbols we add. The new set of consistant formulas is still consistant in the new language $\mathscr{L}'$.
- Step-3: We further construct the set Δ which contains $\Gamma \cup \Theta$ as a subset and satisfies the property that if there's a deduction sequence in Δ then we the formula is in Δ.
- Step-4: Let A to be the set of all terms from $\mathscr{L}'$ then define an equivalence relation E on A:

  $$x \simeq y \text{ iff } (x = y) \in \Delta$$

  , check this is actually an congruence relation.
- Step-5: We create the structure which has domain A/E, since E is a congruence relation we know that A/E is well-defined.
- Step-6: Check this is the structure we want to prove the completeness theorem.