A CONCISE INTRODUCTION TO MATHEMATICAL LOGIC

Wolfgang Rautenberg

December 2, 2024

Contents

1 Propositional Logic

Overview:

Boolean functions and boolean formula

Propositional valuation
Realization $\omega : \mathrm{PV} \to \{0,1\}$
Propositional Model

Boolean Function: $f : \{0,1\}^n \to \{0,1\}$

By recursive definition of **Formulas**, extend ω to:
$\bar{\omega} : \varphi \to \{0,1\}$, $\varphi \in F_n$.

$\bar{\omega}\varphi = f(\omega\vec{p})$

Fact:
Assign a truth value to a formula $\Leftrightarrow$ Assign a true value at all occurance of the PV

Semantic of Propositional Logic

Semantic Equivalence:
$\alpha \equiv \beta \Leftrightarrow \omega\alpha \equiv \omega\beta$

Motivation:
If every boolean functions can
be represented by a boolean formula?

Replacement Theorem:
$\alpha \equiv \alpha' \Rightarrow \varphi \equiv \varphi'$,
φ' is obtained from φ by replacing occurence of α in φ, by α'.

Every boolean function f can be represented by DNF α_f, i.e.
$\alpha_f := \vee_{f\vec{x}=1} p_1^{x_1} \wedge \cdots \wedge p_n^{x_n}$,
where $(x_1, \dots, x_n) \in \{0,1\}^n$, $p_i^0 := p_i, p_i^1 := \neg p_i$.

Functional Complete:
Logical signature is functional complete it can represent every boolean functions.

Logical Consequence and its properties

Model:
If $\omega\alpha = 1$, ω is a model of α,
or ω satisfies α, denote as $\omega \vDash \alpha$.
This definition can naturally be extended to
a set of formulas X.
$\vDash$ is the satisfiability relation.
$\alpha(X)$ is satisfiable if there exists a model.

Tautology:
If all ω satisfy α,
α is a tautology.
Contradiction:
If all ω don't satisfy α,
α is a contradiction.
Negation of a tautology isn't a contradiction.

Logical Consequence:
If $\omega \vDash X$, then $\omega \vDash \alpha$.
We say α is a logical consequence of X.

Basic properties of $\vDash$:
(R) : **Reflexive**
(M) : **Monotone**
(T) : **Transitive**

Fact of Tautology:
Tautologies of the form $\alpha \vee \neg\alpha$ is implied by $p \vee \neg p$.

(S) : **Invariance Substitution**

(F) : **Finitary:**
$X \vDash \alpha \Rightarrow X_0 \vDash \alpha$,
$X_0 \subseteq X$, X_0 finite.

$\vDash$ shares the properties (RMTS) with almost all classical / non classical Logical systems.
A propositional consequence relation $\vdash$, is a relation
between sets of formulas and formulas of a given FOL $\mathcal{F}$,
with properties corresponds to (RMTS).

Deduction Theorem:
$X, \alpha \vDash \beta \Rightarrow X \vDash \alpha \to \beta$.

Syntax of Complete Calculus for $\vDash$:

Derivability Relation $\vdash$:

$\vdash$ is a relation between set of formulas and formulas.
If $\vdash$ can applies to pair (X, α), denote as $X \vdash \alpha$,
call α is a derivation from X. Otherwise $X \nvdash \alpha$.
(X, α) is called **sequent** w.r.t. Gentzen.

Calculus on $\vdash$:

(1) : A functional complete logical signature $\{\wedge, \neg\}$;
(2) : 6 basic rules which are designed for completeness,
these 6 rules will be showed in the equivalent definition below;
(3) : Provable rules / Derivable rules are the rules
can be intefere from (1) (2).

Remark: $\frac{X, \neg\alpha \vdash \beta, \neg\beta}{X \vdash \neg\alpha}$.

Derivability Relation $\vdash$ (Equivalent Definition):

Smallest relation $\subseteq \mathfrak{B}\mathscr{F} \times \mathscr{F}$ and
closed under the following 6 rules:

$$\frac{}{\alpha \vdash \alpha} \qquad \frac{X \vdash \alpha}{X' \vdash \alpha}, X \subseteq X'.$$
$$\frac{X \vdash \alpha, \beta}{X \vdash \alpha \wedge \beta} \qquad \frac{X \vdash \alpha \wedge \beta}{X \vdash \alpha, \beta}$$
$$\frac{X \vdash \neg\alpha, \alpha}{X \vdash \beta} \qquad \frac{\alpha, X \vdash \beta | X, \neg\alpha \vdash \beta}{X \vdash \beta}$$

Conventions of $\vdash$:

$X, \alpha \vdash \alpha \Leftrightarrow X \cup \alpha \vdash \alpha$;
$X \vdash \alpha, \beta \Leftrightarrow X \vdash \alpha \text{and} X \vdash \beta$;
The syntax is of the form $\frac{\text{Premises}}{\text{Inference}}$.

$\xrightarrow{\text{identical with}}$ **Consequence Relation $\vDash$**

Syntactical Meaning of $X \vdash \alpha$:

(X, α) can be obtain
from stepwise application.

Derivation:

Derivation is the records of the
stepwise application process.

Derivation(Formal):

A derivation of (X, α)
is a tuple $(S_0, \dots, S_{n-1}, S_n)$
where $S_n = (X, \alpha)$, each of S_i
is obtained by the following rules:
(1) IS ;
(2) Basic rules applies on $S_k, k \leq i$.

Semantics of $\vdash$:

Property of sequents:
$\mathcal{E}$ is a property of sequents,
i.e. $\mathcal{E}$ can apply on the pair (X, α).
Induction on property:
Let $\mathcal{E}$ be a property closed under $\vdash$,
then $X \vdash \alpha$ implies $\mathcal{E}(X, \alpha)$.
$\mathcal{E} := \vDash$ is a good example.

$\xrightarrow{\text{induce}}$ **Soundness(Semantic):**
$\vdash \subseteq \vDash$

With induction on property, we will deduce
a symmetric process
which builds a relation of $\vdash$ and $\vDash$.

Finiteness theorem for $\vdash$:
If $X \vdash \alpha$, then exists finite $X_0 \subseteq X$ with $X_0 \vdash \alpha$.

Consistent:
$X \subseteq \mathcal{F}$ is consistent if $X \vdash \alpha$
for all α; else consistent.
$X \subseteq \mathcal{F}$ is maximal consistent
if X is consistent and any
$X \subseteq Y$, is inconsistent.

Lindenbaum's theorem:
Every consistent set X can be extended
to a maximally consistent set $X' \supseteq X$

$C^+ : X \vdash \alpha \Leftrightarrow X, \neg\alpha \vdash \bot$;
$C^- : X \vdash \neg\alpha \Leftrightarrow X, \alpha \vdash \bot$.

Properties of maximal consistent:
(1) : $X \vdash \neg\alpha \Leftrightarrow X \nvdash \alpha$;
(2) : Maximally consistent set X is satisfiable.

Completeness theorem:
$X \vdash \alpha \Leftrightarrow X \vDash \alpha$.

Results of completeness theorem:
(1) : If $X \vdash \alpha$, then fintite $X_0 \subseteq X$, $X_0 \vDash \alpha$.
(2) : A set X is satisfiable then each finite subset of X is satisfiable.

Hilbert Calculi:

Hilbert Calculi $\vdash_H$:
(1) Logical Signature : $\{\neg, \wedge\}$
(2) Logical axiom scheme :
1. $(\alpha \to \beta \to \gamma) \to (\alpha \to \beta) \to \alpha \to \gamma$;
2. $\alpha \to \beta \to \alpha \wedge \beta$;
3. $\alpha \wedge \beta \to \alpha$, $\alpha \wedge \beta \to \beta$;
4. $(\alpha \to \neg\beta) \to \beta \to \neg\alpha$.

1.1 Boolean Functions and Formulas

1.1.1 What is Prositional Language?

Definition 1.1 (n-ary Boolean Functions)**.** $f : \{0,1\}^n \to \{0,1\}$ is called an n-ary Boolean function or Truth function.

Proposition 1.2. *n-ary Boolean function f has 2^n tuples in its **domain**, this gives 2^{2^n} ways to construct an n-ary Boolean function.*

Definition 1.3. We denote the totality above as $\mathbf{B}_n$, that is $\mathbf{B}_n = 2^{2^n}$.

Definition 1.4 (Propositional Language(defined by induction))**.** Given a set of logic symbols(*Logical signature*) and a set of variables. We define propositional language $\mathfrak{F}$ inductively:

1. one-element strings are formulas(*Prime Formulas*).
2. If α, β are all formulas then $\alpha \circ \beta$ here $\circ$ refers to binary Boolean Functions and $\neg\alpha \in S$.

Based on set theory we also have another definition:

Definition 1.5 (Propositional Language(defined in set-theoretical way))**.** Propositional Language $\mathfrak{F}$ is the smallest set of all **String** S built from *logic symbols* and *propsitional variables*, satisfying the following properties:

f1 $p_1, \cdots \in S$.

f2 $\alpha, \beta \in S$ they are closed under the binary Boolean function and unary Boolean function.

Definition 1.6 (Formula)**.** A string in a propositional language is a formula.

Example 1.1. For Boolean Signature we have the Boolean formulas.

Worth noticing that we don't use parentheses for the unary operation.
Now we let $\mathfrak{F}$ to be the set of all Boolean formulas.
For convention we obey the following rules:

1. Omit the outside parentheses.
2. If order of the logic connectives makes the formula no ambiguity without parentheses then we omit the parentheses.
3. Multiple use of $\to$ we associate to the right; multiple use of other binary connectives we associate them to the left.

In arithmetic one used to associate to the left but x^{y^z} is an example of associate to the right.

Theorem 1.7 (Induction principle for formulas)**.** *Let* E *be a property of strings(We write* $\mathrm{E}\psi$ *to represent* E *is a property of string* ψ*), such that(If one can show):*

- $\mathrm{E}\pi$ *for prime formual* π*,*
- $\mathrm{E}\alpha, \mathrm{E}\beta$*, then the formulas building from* α, β *also have this property.*

Then E *holds for all formula.*

Some Language notation:

- **Such that means suppose to show, means the goal is...**

- **Compound is the words from chemestry, which means that many different kinds of elements together form sth, this sth is so-called compound.**

Theorem 1.8 (Unique reconstruction property)**.** *If α, β where $\alpha \circ \beta$ construct φ, then α, β are uniquely determined.*

This property looks very weird since it somehow have the idea of the free generation and have some idea of unique readability theorem.

Definition 1.9 (Inductive defintion of subformula)**.**

- subformula of prime formula is itself
- subformula of $\neg\alpha$ is $\{\neg\alpha\}\cup$ subformula of α
- for boolean signature just the natural way: itself and subformulas of the component.

By arithmetic one emphersize the numbers and the operations on it.

Definition 1.10 (Propositional valuation(Realization of propositional model))**.** A propositional valuation ω is a function $\omega : PV \to \{0, 1\}$.

Stipulation: a rule must be follow or sth must be done.
we can extend the valuation in natural inductive way:

- $\omega\alpha \circ \beta = \omega\alpha \circ \omega\beta$
- $\omega\neg\alpha = \neg\omega\alpha$

By the extension we can talk about the valuation of formula φ.

For the next part we will talk about the connection between boolean functions and boolean formulas

1.1.2 Correspondence between boolean formulas and boolean functions

1.2 Semantic Equivalence and Normal Forms

2 Predicate Logic

2.1 Overview of the basical objects

Extralogical Structure L:
Constant symbols, function symbols, relation symbols.
e.g. in group theory $\{\circ, e\}$.
L**- Structure** $\mathcal{A}$**:**
$(A, L^{\mathcal{A}})$, A is the domain of $\mathcal{A}$,
$L^{\mathcal{A}}$ consists of the interpretation of the extralogical structure.

$\downarrow$

Alphabet of L:
The alphabet of L consists of L,
logical symbols and varibales. **Strings of L:**
$\mathcal{S} := \{$The set of all strings from $L\}$.

$\downarrow$

Recursive definition of:
Terms
Prime Formulas
Varibales

$\mathcal{L}$:= {The set of all formulas determined by L.}
$\mathcal{L}^0$:= {The set of all the sentences}.
Sentences: Formula with no free variable.

$\downarrow$

L- structure for extralogical symbol L,
naturally become $\mathcal{L}$- structure,
with the recursive definition of terms and formulas.

2.2 Mathematical Structures

Definition 2.1 (”Specific” Structure and related definition)**.** For structure $\mathfrak{A}$ we have the following description:

Notations	A: Domain	relations r, operations f, constant c
Descriptions	finite(infinite) structure	relation(algebraic) structure

- Relation structure has no operation and constant
- Algebraic structure has no relation

We want to study the class of structures, so we need a “connection” which enables us to talk about a class of structures.

Definition 2.2 (Extralogical Signature)**.** A finite set L consisting of relation, operation, and constant symbols of given arity, is a (extralogical) structure.

Definition 2.3 (Closed under operations). $\forall a \in A^n \Rightarrow fa \in A$.

Definition 2.4 (Restriction to a subset of domain). **Restriction for Relation:**

- Intersect product sets

Restriction for Operations:

- Closed under operations w.r.t. the subset which we want to restrict on.

Definition 2.5 (Substructure). Let $\mathfrak{B}$ be an L-Structure, $A \subset B$ nonempty and closed under all operations of B and inherits all the interpratations of constant of B.

Proposition 2.6 (Common properties of binary operations and binary relations). ***Properties for binary operations:***

- *Commutivity;*
- *Associativity;*
- *Idempotent:* $a \circ a = a$*;*
- *Invertible:* $\forall a, b, \exists x, y \Rightarrow a \circ x = b \wedge y \circ a = b$.

Properties for binary relations:

- *reflexive and irreflexive;*
- *symmetric and antisymmetric;*
- *transitive;*
- *connex(Trinity).*

2.3 Syntax of Elementary Languages

Term

1. **Prime Term** : Variables and constants
2. **Compound of functional symbol and terms**

$\longrightarrow$

Formula

1. **Terms**
2. **Compound of relation and terms**
3. **Compound of logical connectives and formulas**

$\downarrow$

1. **Variable**
2. **Free Variable** $\Rightarrow$**Sentences**
3. **Bounded Variable**

Goal: We want to delimit(determine the limit) the theoretical framework which enables us to precisely talk about mathematical structures

This goal arise the definition of object language.

Definition 2.7 (Object Language). Object Language is the language can be described by metalanguage.

Definition 2.8 (Object(w.r.t. object language)). Objects are formalized elements of the language.

To formalize interesting properties of a structure, one need the following things:

Definition 2.9 (Individual Variables(Informal))**.** Individual variables are a “place-holder” with a predicate letter. It stands for unspecified argument of the predicate.

Definition 2.10 (Extralogic Structures(w.r.t. the given language))**.** Sufficient number of relations, funcitons and constant.

The language with the two features above is the first order language or elementary languages, now we give a formal definiton.

Definition 2.11 (First-order language(Informal))**.** First-order language is a set consists of the following type of subsets:

- Alphabet
  - Individual variables(Var): countably many variables.
  - Extralogic structures
- Syntax of first-order logic
- Semantics of first-order logic

Remark:

- **One can only differ two first-order language by the Extralogical Structures.**
- **Individual Variables here often denotes by** $x, y, z, \ldots$**.**

Definition 2.12 (Alphabet)**.** Alphabet is the set of basic symbols of a first-order language determined by a (extralogical) signature L.

3 Gödel's Completeness Theorem

Natural Deduction: Gentzen's type $\vdash_G$
Hilbert Calculi: Modus ponens $\vdash_H$

Gödel's Completeness Theorem
1. $\vdash_H = \vDash$
2. $\vdash_G = \vDash$
3. **Finiteness Theorem:**
$X \vDash \alpha \Rightarrow X_0 \vDash \alpha, X_0 \subseteq X, X_0$ finite.
4. **Compactness Theorem(Syntactic):**
Any set of first order formula X is satisfiable $\Rightarrow$ Every finite subset of X is satisfiable.

A theory is **complete** if it is consistent and has no proper consistent extension.

4 Incompleteness and Undecidability

Overview:

$$
\begin{tikzcd}
\text{Recursive and Primitive Recursive Functions} \arrow[d] & \text{Arithmatic on syntax and formal proofs} \arrow[dl] \\
\text{Representability of recursive functions} \arrow[d] & \\
\text{First Incompleteness Theorem} \arrow[d] \arrow[dr] & \\
\text{Decidability and Undecidability} & \text{First-order arithmetical hierachy}
\end{tikzcd}
$$

Gödel's first incompleteness theorem informal description: **Basic Assumption:**

$$
\begin{tikzcd}
\mathcal{T} : \text{Axiomatic theory} \arrow[r, "\text{describe}"] & \mathcal{A} : \text{Given domain of objects} \\
\mathcal{L} : \text{Language of} \mathcal{T} \arrow[u, "\text{Internal encoding of syntax of} \mathcal{L}"] &
\end{tikzcd}
$$

Result: Sentence γ :"I(γ) am provable in $\mathcal{T}$" is true in $\mathcal{A}$ but unprovable in $\mathcal{T}$

This result is kind of like the liar paradox. "I will not died because of fire." This is true because within the rule I will not die. This is unprovable because we can't make sure the semantic of this sentence within the rules.

4.1 Recursive and Primitive Recursive Functions

Definition 4.1 (Partial Function)**.** Let X, Y be sets, $S \subset X$. $f : S \to Y$ is a partial function from $X \to Y$.

Difference between primitive recursive functions and recursive functions

- Primitive recursive functions are from primitive recursive functions with one input.
- Recursive function: Partial functions take finite tuples of natural numbers and return a single natural number.

Proposition 4.2. *Primitive recursive function $\subset$ Recursive function $\subset$ Partial recusive function.*

Definition 4.3 (Halting Problem)**.**

Goal: Code Undecidability into logic

Definition 4.4 (Computation)**.** Computation is a sequence of configuration of addition mechine.

Remark: Here configuration means the current states.

5 Kappa-categorical

Definition 5.1 (Kappa-categorical)**.** We say a theory is $\aleph_0$-categorical if any countable infinite models are isomorphic.