Proofs and Logic

Hello reader, this is a introductory document for anyone wanting to learn about the logical models of mathematics. Keep in mind I am just a undergrad. A lot of this comes from what I have learned from **Introduction to Proof Theory** by Samuel R.Buss, **Lecture 4: Frege and Extended Frege** by Toniann Pitassi give them a read!.

Contents

I Introduction

Mathematical theories need to be presented in a formal system, in this chapter we will explore different logics that are used or extended in the foundation of mathematics. Due to their frequent adaptation, the details and naming conventions are not what's important but the structure and workings. We will mainly be going over defintions and concepts, but some fun theorems will be presented.

I.1 How to read this document

In the following 3 chapters, we will go over propositional logic, first order logic and higher order logic, exploring each in the same way;

1. First, we will go over the Synthax of the formal language that is associated
2. Then we will look at an interpretation of the Synthax, the Semantics
3. Then we will look at proofs in the domain
4. Finally we might go over some interesting theorems

I.2 Some notes for clairity's sake

This might been covered in previous chapters, however we shall ensure it is said; Logics are the more general systems that have a synthax (well defined purely linguistic form) a sematic (interpretation) and proofs along with their study. Formal systems generally base themselves on a logic and provide axioms in the goal of deriving statements, however at it's basis is just a formalisation of whatever we want to study.

II Zeroth order logic (propositional logic)

This is the simplest, but also the least powerfull (we can't prove or do much with it) that we will see, it is however interresting especially in computer science (see S.A.T problem). This logic bases everything on true and false arguments and binairy connectors.

II.1 Synthax

We define our alphabet as:

- Our variables $V = \{p_1, p_2, ...\}$ for all natural numbers
- Our constants $\top$, and $\bot$ (these are also seen as propositional connectives of arity 0)
- Our propositional connectives with their arity. Often used ones include: $\wedge$ of arity 2, $\vee$ of arity 2, $\neg$ of arity 1, $\rightarrow$ of arity 2, $\leftrightarrow$ of arity 2. There are many more and we can even build all of them from one but that has to do with the Semantics. These are also called logical connectives.

We define the set of Formulas as:

- all elements of V are Formulas.
- $\top$, and $\bot$ are Formulas.
- If $A_1..., A_n$ are Formulas and $*$ is a propositional connective or arity n then $* (A_1..., A_n)$ is too.

This creates a tree like structure.
Here are a couple examples (here they are written in infix notation, $p_1 \wedge p_2$ instead of $\wedge (p_1, p_2)$:
$p_1 \wedge \neg p_2 \qquad \bot \wedge p_4 \qquad p_2 \wedge \neg(p_2 \vee p_1)$

We use the parenthesis for clarity, by convention $\neg$ has most priority and then $\wedge$, followed by $\vee$
> With $x_1..., x_n$ the variables used in a Formula, the formula is essentially a propositional connective of arity n however we shall reserve that name for the ones defined in our alphabet and call it a n arity relation.

We can define the variables in a formula as :

- $\mathrm{Var}(\top) = \emptyset$ and $\mathrm{Var}(\bot) = \emptyset$
- $\mathrm{Var}(p_k) = \{p_k\}$
- $\mathrm{Var}(* (x_1..., x_n)) = \mathrm{Var}(x_1) \cup ... \cup \mathrm{Var}(x_n)$

II.2 Semantics

Here we will offer an interpretation of a formula. We define true and false written T, F. We call a truth assigment τ a function $V \rightarrow \{T, F\}$, which we can limit to be from the variables used in a certain formula ($\mathrm{Var}(A)$) instead of all of V, this gives us the advantage of having a finite set of variables.

We then extend the value of our truth assigment $\tilde{\tau}$ for any Formula by Let A be a fomula

- If $A = p_n$, $\tilde{\tau}(A) = \tau(p_n)$
- If $A = * (p_1..., p_n)$, a propositional connective of arity n we define $\tilde{\tau}(A) = \tilde{*} (\tilde{\tau}(p_1)..., \tilde{\tau}(p_n))$ with $\tilde{*}$ as defined bellow, in what is called a truth table.

A	B	$\wedge$	$\vee$	$\rightarrow$	$\leftrightarrow$
F	F	F	F	T	T
F	T	F	T	T	F
T	F	F	T	F	F
T	T	T	T	T	T

A	$\neg A$
F	T
T	F

$\top$	$\bot$
T	F

We say $\tilde{\tau}$ satisfies a Formula A if $\tilde{\tau}(A)$ is true (T). We say A is satisfiable if there exists such a truth assigment, that truth assigment is called a model for A. We say Γ, a set of Formulas is satisfiable if there exists such a truth assigment that satisfies all of its elements. We say A is a tautology if every truth assigment satisfies A we write $\vDash A$. We say Γ tautologically implies A if for every truth assigment that satisfies Γ, it satisfies A, we then write $\Gamma \vDash A$. Notice that if $p_n \notin \text{Var}(A)$ then $\tilde{\tau}(A)$ is independant of $\tilde{\tau}(p_n)$.

Note on functional completeness: We say a set of logical connectives is functionnaly complete if we can build every truth table with them. $\{\rightarrow, \wedge, \vee, \neg, \leftrightarrow\}$ is functionaly complete, in fact we can limit ourselves to a single connective: nand $(a, b : \neg(a \wedge b))$ or a nor $(a, b : \neg(a \vee b))$.

A	B	$\bar{\wedge}$ (nand)	$\bar{\vee}$ (nor)
F	F	T	T
F	T	T	F
T	F	T	F
T	T	F	F

II.3 Proofs

Say we have a Formula A using the variables $p_1 ..., p_n$ to figure out if it is a tautology, we can just test out all possibilities on say a computer. Thats "just" 2^n possibilities, sometimes acceptable. However, not only does it quickly get out of hand, we will also learn a method that will help us furter on, when we extend it to higher orders of logic. Proofs.

Here we will introduce some methods of prooving that we can show are both complete (if it's a tautology we can proove it) and sound (if it verifies the proof it's a tautology).
Proof theory studies how these systems interact, if they can prove the same things and the relation between the size of the proofs.

II.3.1 Substitutions, sequents, axioms and inference rules

In order to prepare for proofs we'll define some notions first.
With P and Q formulas and p a variable, we define the substitution of p with Q within P ($P[Q/p]$) as:

- If $P = p_k \in V$, $P[Q/p]$ is Q if p_k is p else p_k
- If $P = *(x_1 ..., x_n)$, $P[Q/p]$ is $* \left(x_1[Q/p] ..., x_{n[Q/p]}\right)$

For example $(\neg p_1 \wedge (p_2 \vee p_1))[(p_3 \vee p_1)/x]$ is $(\neg(p_3 \vee p_1) \wedge (p_2 \vee (p_3 \vee p_1)))$

Inference rules are descriptions of how we will pass from one formula to another, written under the form $\frac{P_1 ..., P_n}{\tilde{P}}$ with $P_1 ..., P_n$ and $\tilde{P}$ formulas where the top formulas are previously deduced

formulas and the bottom one is the one deduced. One famous example (probably the most used) is *modus ponens* $\frac{P_1, P_1 \to P_2}{P_2}$. In proofs these are used to pass from one line to another.
Axioms are a little different, they are used along with inference rules, they are formulas, along with all of their possible substitutions, for example $(\varphi \wedge \psi) \to \varphi$ where φ and ψ denote all possible formulas. These are used within inference rules within the proof.

For example with modus ponens as the only inference rule and $\neg(p_1 \wedge p_2) \to \neg p_1 \vee \neg p_2$, $\neg\neg p_1 \vee p_2 \to p_1 \vee p_2$ and $\neg(p_1 \wedge \neg p_1)$ as our axioms, we can show the law of excluded middle $(p_1 \vee \neg p_1)$.
1. $\neg(p_1 \wedge \neg p_1)$ (axiom 3)
2. $\neg\neg p_1 \vee \neg p_1$ (first line + axiom 1 + modus ponens)
3. $\neg\neg p_1 \vee \neg p_1 \to p_1 \vee \neg p_1$ (axiom 2)
4. $p_1 \vee \neg p_1$ (last two lines + modus ponens)

Finaly, sequents are a little like a mix between both, they are the main element in a different type of proof system called sequent calculus. They have both roles, they express things that are true $p \to p$ while also giving us a way to pass between lines of the proof.
They have a very specific form, to write them we use a character that means "implies" called the sequent arrow, but for clarity reasons is syntaxicaly different then our logical connective $\to$, it is often $\vdash$ or $\supset$, here we will use $\vdash$. Sequents are under the form $P_1..., P_n \vdash \widetilde{P_1}..., \widetilde{P_k}$ which should be interpreted as $P_1 \wedge ... \wedge P_n \to \widetilde{P_1} \vee ... \vee \widetilde{P_k}$. $P_1..., P_n$ are called the antecedent and $\widetilde{P_1}..., \widetilde{P_k}$ is called the succedent, they are both sedents. In sequent calculus the inference rules are all under the form of $\frac{\Gamma}{\Delta}$ where Γ and Δ are sequents. For example:

$$\frac{}{p_1 \to p_1} \qquad \frac{p_1..., p_k \to \tilde{p}_1..., \tilde{p}_n}{p_1..., p_k, p_{k+1} \to \tilde{p}_1..., \tilde{p}_n}$$

There are also several ways to look at the sedents, such as in the form of a set of formulas, in that sense the rules of exchange ($\frac{..., a, b, ... \to c}{..., b, a, ... \to c}$) are natural, while in other perspectives, they may be presented as inference rules, furthermore there may be several ways to express things like $..., a, b, ...$ such as $..., A, B...$ where A and B are every possible formula, or Γ, a, b, Δ where Γ and Δ are cedents (our cedent Δ, a, b, Γ would then be $\Delta \cup \{a, b\} \cup \Delta$).

II.3.2 Frege and Hilbert style proof system

Frege and Hilbert style proofs are general categories of systems that share usefull properties, Frege sytems are a subset of Hilbert systems, however these systems are equivalent in what they can prove, the difference is mainly minimalism compared to readability.
Frege systems use axioms and Defintion: A frege style prooving system consists of a finite set of inference rules under the form $\frac{p_1..., p_n}{\tilde{p}}$, and a set of functionaly complete logical connectives, where proofs created by applying the rules of deduction with any substitutions is a sound and implicationally complete system.
This might seem a little vague, let's look at some examples.

II.3.3 Hilbert style

In Hilbert style proofs, there are a set of axioms under the form

$$\frac{P_1(p_1..., p_n)..., P_{k(p_1..., p_n)}}{\widetilde{P(p_1..., p_n)}}$$

With $P_1 ..., P_k, \tilde{P}$ formulas of arity n. Then each step in the proof is a line written from previous lines, by substituting the axioms.

For example :

$$\frac{x_1}{x_1 \vee x_2}, \frac{x_1 \vee x_1}{x_1}, \frac{x_1 \vee (x_2 \vee x_3)}{(x_1 \vee x_2) \vee x_3}, \frac{x_1 \vee x_2, \neg x_1 \vee x_3}{x_2 \vee x_3}, \frac{}{x_1 \vee \neg x_1}$$

Example Proof: We will show the law of excluded middle ($p \vee \neg p$) from the following inference rules :

$$a)\frac{}{\neg(p_1 \wedge \neg p_1)}, b)\frac{\neg(p_1 \wedge p_2)}{\neg p_1 \vee \neg p_2}, c)\frac{\neg\neg p_1}{p_1}, d)\frac{\neg\neg p_1 \vee p_2}{p_1 \vee p_2}, e)\frac{p_1 \vee p_2}{p_2 \vee p_1}$$

Followed by a couple more to make the system complete. A proof is as follows (what is in [] is not part of the proof, just explanation):
1. $\neg(p_1 \wedge \neg p_1)$ [using axiom a)]
2. $\neg p_1 \vee \neg\neg p_1$ [using line 1), and axiom b) by substituting p_2 with $\neg p_1$]
3. $\neg\neg p_1 \vee \neg p_1$ [using line 2) and axiom e) substituting p_1 with $\neg p_1$ and p_2 with $\neg\neg p_1$]
4. $p_1 \vee \neg p_1$ [using line 3) and axiom d) substituting p_2 with $\neg p_1$]

Axioms need not be used, for example c).

Modus Ponens: We will point out a very used rule, *modus ponens* that states $\frac{p_1, p_1 \to p_2}{p_2}$.

II.3.4 Gentzen style (or PK sequent calculus)

Gentzen style is a little diffrent to Hilbert, it is a tree like structure of sequents.

II.4 Theorems

II.4.1 Canonical form

This is a way to normalise formulas, in order to simplify encoding for solving with computers (or without).

minterm: Is a conjunction ($\wedge$) of n variables such that for each variable x_k in $x_1 ..., x_n$, x_k or $\neg x_k$ is used. For example $x_1 \wedge x_2 \wedge \neg x_3 \wedge x_4$ (aka a product term in which each variable appears once)
maxterm: Is a disjuction ($\vee$) of n variables such that for each variable x_k in $x_1 ..., x_n$, x_k or $\neg x_k$ is used. For example $x_1 \vee x_2 \vee \neg x_3 \vee x_4$ (aka a sum term in which each variable appears once)
minterm canonical form (SoP (or sum of products)): Is a sum term of minterms.
maxterm canonical form (PoS (or product of sums)): Is a product term of maxterms.

Each formula of arity n as an equivalent maxterm and minterm of arity n.

II.4.2 SAT Problem (Satisfiability)

The SAT problem is, given a formula of arity n, can we find an interpretation that satisfies it. This is NP complete. It is interesting to note for historical reasons that this is the first reference (or at least first proof of) and NP complete problem.

III First order logic (quantification or predicate logic)

This is one of the most frequent logic systems and incredibly powerfull, it remains complete and sound but it is much more flexible than zeroth order logic. It can also be seen as an extension based on quantification.

III.1 Synthax

Our alphabet consists of:

- logical symbols
  - variables $p_1, p_2, ...$ indexed by $\mathbb{N}^*$
  - logical connectives from zeroth order logic (a functionally complete set)
  - the qantifiers: $\forall$ and $\exists$
  - an optional $=$ symbol
- non-logical symbols
  - predicate (aka relation) symbols, for each arity n there are an infinity indexed by $\mathbb{N}^*$: $R_1^n, R_2^n, ...$
  - function symbols, for each arity n there are an infinity indexed by $\mathbb{N}^*$: $f_1^n, f_2^n, ...$, note that functions of arity 0 are called constants

We define
- Terms:
  - all variables are terms
  - if f_k^n is an n arity function and $t_1, ...t_n$ are terms $f_k^{n(t_1,...t_n)}$ is a term
- Formulas (aka well formed formulas)
  - if P_k^n is an n arity relation and $t_1, ...t_n$ are terms $P_k^{n(t_1,...t_n)}$ is a formula
  - if using $=$ and t_1, t_2 are terms then $t_1 = t_2$ is a formula
  - if $*$ is an n arity logical connecive and $\alpha_1, ...\alpha_n$ are formulas then $* (\alpha_1, ...\alpha_n)$ is a formula (though oft in infix notation)
  - if φ is a formula and x is a variable then $\forall x.\varphi$ and $\exists x.\varphi$ are formulas (the . is oft ommited or parenthesis are used)

For example, in peano arithmetic, we have two functions, S, the successor function of arity 1, and 0 of arity 0. In ZF we have $\in$ as a relation. It is important to note that the notion of funciton and relation here is one level above the objects in the universe, we might not be able to construct the sets of all sets but we can construct a relation over all sets without a problem. Note also that functions can be built as relations with the added axiom that there is at most one (from the point of view of $=$) second value for all first values.

III.1.1 Free and bound variables:

For substitutions within a formula we first define free and bound variables inductively. It must be noted that

Free variables:
- $\text{Free}(p_n) = \{p_n\}$
- $\text{Free}(\top) = \emptyset$ and $\text{Free}(\bot) = \emptyset$
- $\text{Free}(* (x_1, ...x_n)) = \text{Free}(x_1) \cup ... \cup \text{Free}(x_n)$

- $\text{Free}(\forall pP) = \text{Free}(P) \setminus \{p\}$ and $\text{Free}(\exists pP) = \text{Free}(P) \setminus \{p\}$

Bound variables:
- $\text{Bound}(p_n) = \emptyset$
- $\text{Bound}(\top) = \emptyset$ and $\text{Bound}(\bot) = \emptyset$
- $\text{Bound}(* (x_1, ...x_n)) = \text{Bound}(x_1) \cup ... \cup \text{Bound}(x_n)$
- $\text{Bound}(\forall pP) = \text{Bound}(P) \cup \{p\}$ and $\text{Bound}(\exists pP) = \text{Bound}(P) \cup \{p\}$

III.2 Semantics

Next we will define an interpretation for out logic. We define the (nonempty) universe Ω the set of all objects we work on, for each function of arity n used we define a function from Ω^n to Ω, and for each relation symbol of arity n we define a relation over Ω (a subset of Ω^n). We also define truth tables for our logical connectives.
Please note that the $=$ symbol is usualy considered equality over the universe, however there is an alternative definition as a relation with the following axioms:
- Reflexivity: $\forall x.x = x$
- Symmetry: $\forall x \forall y.x = y \leftrightarrow y = x$
- Transitivity: $\forall x \forall y \forall z.x = y \wedge y = z \rightarrow x = z$
- Substitution: with R a n arity relation $\forall x_1 ... \forall x_n \forall y_1 ... \forall y_n.x_1 = y_1 \wedge ... \wedge x_n = y_n \rightarrow (R(x_1..., x_n) \leftrightarrow R(y_1..., y_n))$ and with f a n arity function $\forall x_1 ... \forall x_n \forall y_1 ... \forall y_n.x_1 = y_1 \wedge ... \wedge x_n = y_n \rightarrow (f(x_1..., x_n) = f(y_1..., y_n))$.

Furthermore constants like $\top$ or $\bot$ can be represented by a function of arity 0.
We can now define the truth function of formulas for first order logic, inductively much like prepositional logic.
- With R a n arity relation and $t_1..., t_n$ terms then $\tau(R(t_1..., t_n))$ is True if $(t_1..., t_n)$ is in the set associated to P else False.
- With $*$ a n arity logical connective we do as prepositional logic
- With P a formula, $\forall p.P$ is True if $\tau(P)$ is true for p taking any possible value in Ω else false, and $\exists p.P$ is true if there exists a value in Ω p could take such that $\tau(P)$ is True, else False

III.3 Proofs

Proofs in first order logic are usually extended versions of prepositional logic with the addition of axioms handling $\forall$ and $\exists$ such as:

$$P(p_1) \rightarrow \exists p_2 P(p_2), \qquad \forall p_1 P(p_1) \rightarrow P(p_2)$$

IV Second order logic

This will be added later on

V Higher order logic (aka Simple type theory)

This will be added later or in the typw theory segment