Course: Set Theory(A formal language)

Sebastián Caballero

December 14, 2022

A simple notes template. Inspired by Tufte-LaTeXclass and beautiful notes by

`https://github.com/abrandenberger/course-notes`

1 The need for a formal language

In our study for set theory, we want to develop a formal basis for mathematics. The intuition about why sets are chosen over other elements like numbers, is because they let us define structures in an easy way that numbers don't let us, and even they can be described in terms of sets. But before we develop the theory, we need to give a framework in which express our ideas.

During this first part, we will develop a bit of a formal language that supplies our needs since for a more extensive treatment, in a long future I will study Logic and Model theory.

2 Development of the formal language

2.1 A few rules before start

In the start for our language, we manage a *metalanguage*. In our case, it is the english, but any common human language is okay. We are not going to explore the rules for the metalanguage, rather we will construct rules for our formal language from here.

First, we need to distinguish that in math, there are objects and relations between them. We define *terms* to be names and symbols that describe specific objects like numbers, structures, descriptions and definitions. Second, the well formed formulas describe relations between the terms and are also described through symbols or descriptions. From here, let's develop some important definitions.

Before giving a complete development of the well formed formulas, we need to specify the scope of the symbols for our language. What we are creating is called *a language of first order*, in which there are two types of symbols: logic symbols that are accepted in every theory constructed with the language and never change; and the specific symbols, that are relative to each theory and can be interpreted differently.

2.2 The symbols

So, let's start by defining what will be our logic symbols for every theory we develop.

> **Definition 2.1** (Logic symbols). *For our language, we define the logic symbols that follows:*
>
> - *The parenthesis "(" and ")" which are grouping symbols*
> - *The logic connectives* $\wedge, \vee, \rightarrow, \leftrightarrow, \neg$
> - *Variables, which are represented as letters in the latin and greek alphabet and sometimes can be written with subindex or super index.*
> - *The quantifiers* $\forall$ *and* $\exists$*(Though the quantifier* $\exists$ *can be written as* $\neg(\forall x)(p(x)))$
> - *The equality symbol* $=$

These symbols can be used in every theory created with them, and since we are hoping to put the foundations of math in set theory, described with this language, they are valid in almost all math. Now, let's define what should be the specific symbols:

> **Definition 2.2** (Specific symbols). *In each theory created with the language, we can have three types of specific symbols:*
>
> 1. ***Constant symbols:*** *Represent specific objects within the theory and it's meaning never change. At the start, a theory can dispense with them.*
> 2. ***Predicate symbols:*** *They describe properties of the objects or relations between them. For a relation or description of n objects, we can describe predicates of n arguments.*
> 3. ***Functional symbols:*** *They can describe operations between symbols and objects, and like the predicate symbols, an operation of n objects is called a function of n arguments.*

For example, if you are working with geometry, a constant symbol could be the plane $\mathfrak{P}$ and if you are working in the real numbers, they can be 1 and 0. Predicate expressions could be T is a triangle or $x < y$, and functional symbols could be $x + y$ and $A \cap B$.

2.3 *Syntactical rules*

We are going to define a classification for the symbols and rules that create expressions we can use in the theory. The will let us develop a sufficient formal way to describe set theory.

Definition 2.3 (Terms). *The variables and constant symbols are terms in the language. if f is a functional symbol of n arguments and $t_1, t_2, \ldots t_n$ are terms, then $f(t_1, t_2, \ldots, t_n)$ is also term in the theory.*

With this, expressions like 1, 0, $1 + 0$, $1 \cdot 0$ and so on are terms. In set theory for example, $\emptyset$ and $A \times B$ could be now terms. Now, we will define how to combine them in well formed formulas.

Definition 2.4 (Well formed formulas and atomic formulas). *If R is a predicate symbol of n arguments and $t_1, t_2, \ldots, t_n$ are terms, then $R(t_1, t_2, \ldots, t_n)$ is an atomic formula. Also, if t_1 and t_2 are terms, then $t_1 = t_2$ is also an atomic formulas. From that, we state that:*

1. *Every atomic formula is a w.f.f*
2. *If α is w.f.f then $(\neg\alpha)$ is so too*
3. *If α, β are w.f.f then $(\alpha \wedge \beta), (\alpha \vee \beta), (\alpha \rightarrow \beta)$ and $(\alpha \leftrightarrow \beta)$ are w.f.f too.*
4. *If α is w.f.f then $(\forall x)(\alpha)$ and $(\exists x)(\alpha)$ is also a w.f.f.*
5. *Any finite combination of the previous four rules create a w.f.f*

So, we are just done with this part of the construction of the language. We have rules, and we have just established what is valid in the language. But there is just one thing missing, and it is whenever we are expressing a true statement or not. And this will be our next mission.

3 *Symbolic Logic*

With our terms well defined, we can create now what is called symbolic logic. We are seeking for the math to be a logic system in which we can develop in a logic way our results. This time, we are going to be a bit less formal about the details, since we are moving in the terrain of pure mathematical logic and philosophy.

Definition 3.1 (Proposition)**.** *A proposition is a w.f.f that can be derived from the rules* $1,2,3,5$ *that we gave in the last section. If a w.f.f is just an atomic formula, it is called an atomic proposition, otherwise, it is a compound proposition*

With this is mind, we want to explain that certain propositions are always true. For example *Be or not to be*. With this, we will define that a proposition will always have exactly one truth value: True or False. From this, we define assignment rules about the logic connectives and their truth values.

Definition 3.2 (Truth values)**.** *If P and Q are propositions, then it holds that*

1. *If P is False, then $\neg P$ is True. If P is True then $\neg P$ is False*
2. *$P \wedge Q$ is True if and only if P is True and Q is True*
3. *$P \vee Q$ is False if and only if P is False and Q is False*
4. *$P \rightarrow Q$ is False if and only if P is True and Q is False*
5. *$P \leftrightarrow Q$ is True if and only if P and Q has the same truth value*

Now, since P and Q, and even more propositions can have many combination of truth values, we make an structure called truth table that let us see the truth values of propositions. For example, for $P \wedge Q$ it would be:

P	Q	$P \wedge Q$
T	T	T
T	F	F
F	T	F
F	F	F

where T represents True and F represents False. The next definition is important for the development of mathematics and proofs

Definition 3.3 (Tautology, Implication and Equivalence)**.** *Let P and Q be propositions. We say that*

1. *P is a tautology if and only if P is an atomic proposition whose truth value is True or if it is a compound proposition that no matter the truth values of its atomic proposition, its true value is True*
2. *P implies Q if the proposition $P \rightarrow Q$ is a tautology. We denote it by $P \Rightarrow Q$*
3. *P is equivalence to Q if the proposition $P \leftrightarrow Q$ is a tautology. We*

denote it by $P \equiv Q$

It is left as a exercise with truth tables prove that the next proposition are tautology.

Theroem 3.1. *Let P, Q be propositions and denote T and F as propositions that are always true and false respectively. Then*

1. $P \equiv \neg(\neg P)$
2. $P \to Q \equiv \neg Q \to \neg P$
3. $F \to P$
4. $P \vee T$
5. $\neg(P \wedge F)$
6. $\neg P \vee Q \equiv P \to Q$
7. $\neg(P \vee Q) \equiv \neg P \wedge \neg Q$
8. $\neg(P \wedge Q) \equiv \neg P \vee \neg Q$

Also, an important fact used frequently whose proof is far from the scope of this study, is related to the quantifiers.

Proposition 3.1. $\neg(\forall x)(p)$ *is the same as* $(\exists x)(\neg p)$ *and* $\neg(\forall x)(p)$ *is the same as* $(\forall x)(\neg p)$.