

# Theory

This chapter is dedicated to mathematical theory such as Set Theory, Type Theory, Logic and alike

- [First Order Logic](#)

# First Order Logic

## Before we Start

To be frank, this is kind of an off the wall paper. First order logic is not something you would normally come across unless you are doing rather advanced mathematics or AI theory. I have always been fascinated by it and ever since I learned it I feel that my problem solving and deduction skills have improved greatly. I will also admit I do not expect many people to actually read all of this and take away everything, it is a complicated topic and I plan on only scratching the surface. My hope really is to just plant a seed, and if you are someone who lets the seed grow and runs with it then thats great, if not thats fine. I will have fun writing this paper none the less.

## What is First Order Logic (FoL)

In short, it's a refined and strict language that allows you to write logical formulas that allows for both quantifiers and predicates. Already this may seem like a lot but we can break this down. By strict language I specifically mean that it has very strict rules. The syntax is either correct or it is not, down to a computer being able to determine if it is well formed easily. Quantifiers are just ways of saying the scope some given set, and there really are only 2, either ALL OF or AT LEAST ONE OF. I will get into examples of that next. Predicates can be thought of as functions, it takes in a variable and returns a value. Most of the time when you look at simple FoL you will see examples where the predicate is logical, there is no defined function, but rather the word used as the function is the definition. For example `round(ball)` would be true vs `round(square)` is false. We don't need to define the function for `round()` since the reader knows what that means. Now you can define predicates if you wish, that is up to you.

## What does the language look like?

The language can get very messy if you are not prepared for it, and I find a lot of papers and wikis on FoL gloss over the symbols before just shoving formulas down your neck. So lets table out the main ones we will be working with

### Quantifiers

Symbol	Formal Name	Meaning
$\forall$	universal quantification	For All of
$\exists$	existential quantification	There exists

The upside down A and backwards E really are very simple once you get past the fact that they are very abnormal characters. Simply put the  $\forall$  just means that all values it's referring to are considered where as with  $\exists$  only 1 needs to be taken into account. A good example would be  $\forall \text{food tastes good}$  vs  $\exists \text{food tastes good}$ , which is like saying "All food tastes good" verses "Some food tastes good". In the first we are saying all food must taste good for the statement to be true and in the second technically we are saying that only one food needs to taste good for it to be true.

I do want to call out 1 more, we will not be using it but it does exist and it is  $\exists!$  which just means "there exists 1 and only 1". Sticking with our last example of food you could assume a really picky child would say  $\exists! \text{food tastes good}$  and they could be referring to like chocolate or something. Not the best example but you get the point.

## Predicates

Predicates are evaluations we make to objects. As I stated before, and for this paper, we will stick to using words that already have meaning to be our predicates. Using our last example we can just turn  $\text{tastes good}$  into a predicate by treating it like a function, for example  $\forall \text{food goodTaste}(\text{food})$  vs  $\exists \text{food goodTaste}(\text{food})$ . Now we can start to look at this as a computable test in that for the  $\forall$  we could loop through all foods and if we find a single one where  $\text{goodTaste}(\text{food})$  returns false it is then false where as with  $\exists$  we could loop through all food and if a single one returns true for the same function then the statement is true.

## Logical Connectives

This is the first time I brought these up but they are used in every language from speech to computers. They are your ANDs, ORs, and NOTs for the most part. Since this is a well defined language they had to make it fun and use their own symbols for Everything so lets table them out

Symbol	Meaning
$\neg$	NOT
$\wedge$	AND
$\vee$	OR

There are more and you can see them [here](#) but for what we will be doing these are all we need.

## Some extra symbols

I plan on doing a whole paper on set theory as well some day, but basically it is the study of sets and it also has its own set of symbols. I will be using one main one from this language to help clear up what a specific variable is representing which is the "set membership" symbol of  $\in$ . A quick way to see this in action would be with small sets of integers.  $1 \in \{1,2,3\}$  would be true where as  $4 \in \{1,2,3\}$  would be false. I could also say things like  $4 \in \text{positiveIntegers}$  which would then be true. If you are interested you can see a list of symbols [here](#)

# Ok so now what?

Well now that we have some clarification we can start to play a bit and really my favorite is the "beetles and bugs" saying, "All beetles are bugs but not all bugs are beetles". As an adult its kind of trivial but we can use this as a simple example to write some functions. Let's break it down

1. All Beetles are Bugs
  - so we can write this as
  - $\forall \text{beetles} \in \text{bugs}$
2. Not All Bugs are Beetles
  - And we can write this as
  - $\neg \forall \text{bugs} \in \text{beetles}$

Now we can mix things around and try to ensure the statement is true. For example if  $\neg \forall \text{bugs} \in \text{beetles}$  is true then is  $\neg \exists \text{bugs} \in \text{beetles}$  true? No, because there are some bugs that are beetles, so the statements  $\forall \text{beetles} \in \text{bugs}$  and  $\exists \text{bugs} \in \text{beetles}$  are both true. Before we move on I am going to rewrite these in a little different way so it lines up with our next examples.  $\forall \text{beetles} \in \text{bugs}$  can be rewritten like this  $\forall x \text{ bug}(x), x \in \text{set of all beetles}$ , and  $\exists \text{bugs} \in \text{beetles}$  can be rewritten as  $\exists x \text{ beetle}(x), x \in \text{set of all bugs}$ . So what did here was gave a predicate for beetle and bug, basically a function that will determine if the object satisfies the condition of being one or the other. We can say these in a sentence like this  $\forall x \text{ bug}(x), x \in \text{set of all beetles}$  "For all of x, x is satisfied by the predicate bug(), where x is a member of the set of all beetles". Where as  $\exists x \text{ beetle}(x), x \in \text{set of all bugs}$  would be "There exists an x that is satisfied by the predicate beetle(), where x is a member of the set of all bugs".

## How about a mental challenge

Now that you have a taste and know whats being said with the language let's try a simple challenge.

- Given the statements below, which are true
  1.  $\forall x \exists y (x + y = 10), x, y \in \text{Positive Integers}$
  2.  $\forall x \exists y (x + y = 10), x, y \in \text{Integers}$
  3.  $\exists x \exists y (x + y = 10), x, y \in \text{Positive Integers}$
  4.  $\forall x \neg \exists y (x + y = 0), x, y \in \text{Positive Integers}$

In these examples we do not define a predicate name since it is itself a function  $x + y = 10$  in that you can plug numbers into it and check. See if you can figure these out on your own before moving on.

**Click to see Answers**

Now I did throw a curveballs at you here, being that I put 2 quantifiers with 2 variables rather than 1, but when you read it out loud you basically just say AND between the 2

1.  $\forall x \exists y (x + y = 10), x, y \in \text{Positive Integers}$

- False
- In this case we are saying, for all of x there exists a y where  $x + y = 10$ , where x and y belong to the set of all positive integers. Reading it this way does it come off more logically?
- Why is it false?
  - We can plug in a value for x where there is not a single y that could make it true, specifically if " $x > 9$ ". If x was 10 then y would have to be 0 and if x was 11 y would have to be -1 and neither of those are part of the set of all Positive Integers.

2.  $\forall x \exists y (x + y = 10), x, y \in \text{Integers}$

- True
- Why is it True?
  - This one on the other hand is the same BUT we expand our domain out to all integers which would include 0 and negative numbers so now we can find an y for every x. If x was 30, we could put y as -20, if x was -90 we could put 100 for y

3.  $\exists x \exists y (x + y = 10), x, y \in \text{Positive Integers}$

- True
- Why is this True?
  - Now this one has 2  $\exists$  quantifiers which basically tells us that we really only need to find 1 example thats true. We can put 5 for x and 5 for y and boom, mic drop, walk away

4.  $\forall x \neg \exists y (x + y = 0), x, y \in \text{Positive Integers}$

- True
- Why is this True?
  - In this one we added a NOT to the 2nd quantifier and the function changed a little, reading it in english it would be "For all of x there is not a single y that makes  $x + y = 0$  true if x and y are positive integers" and with a little reasoning we can conclude this is a true statement. If we switched the domain to Integers then this statement would be false since we would actually have an infinite number of combinations we could come up with to have  $x + y = 0$ ,  $\{(0,0), (1,-1), (2,-2), \dots\}$

## What is the Takeaway?

If you stuck with me this far that's awesome and really the takeaway is just a new language you can use to refine your deductive reasoning. Basically any statement can be rewritten in FoL and you can use that to really dig into the statement and learn if it's true or not. Or perhaps you need to define more variables or your predicates need to be defined better. I personally have found it

rather rewarding boiling statements down into FoL to really get to an answer, but also Im weird...

Again I have only scratched the surface here, this is a massive and complicated topic and sadly the doorway to it is also complicated and rather esoteric, but based on what you have learned can you understand what my bio line from Lemmy is now?

If you have any questions or want to learn more you can reach me in matrix

@ackermann:hackliberty.org