First Order Logic Syntax

Compared to propositional logic, FOL has the added capacity to refer to individuals with properties and relationships.

Defining $\mathcal{L}$

$\mathcal{L}$ is the symbol representing the set of symbols used in FOL such as

Logical Symbols: connectives ($\vee, \wedge, \rightarrow, \neg, \leftrightarrow$), quantifiers ($\forall, \exists$), free or bound variables (x, u ...), or punctuation (( ) ,)
Non-Logical Symbols: Constants (a, b, c), relation symbols/predicates ($F(), B()$), function symbols ($f(), g()$)

Terms of $\mathcal{L}$

The word "term" is used to refer to either an individual or a variable.
Formally: Term($\mathcal{L}$) = smallest class of expressions of $\mathcal{L}$ closed under the following 3 rules.

1. Every individual symbol a is a term in Term($\mathcal{L}$)
2. Every free variable symbol u is a term in Term($\mathcal{L}$)
3. If $t_1, t_2 \ldots t_n$ $n \geq 1$, are terms in Term($\mathcal{L}$) and f is n-ary function then $f(t_1, \ldots, t_n)$ is a term in Term($\mathcal{L}$)

Basically saying constants, variables and functions applied to terms are terms

Atomic Formuals of $\mathcal{L}$

An atomic formula/atom of $\mathcal{L}$ is the simplest formula that is a complete statement meaning it can be determined true or false.

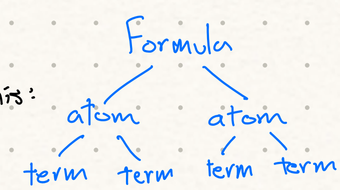
Types of Atoms: 1. A relation applied to terms in the form $F(t_1, t_2 \ldots t_n)$ where F is a relation ($=, <,$ divides ...) and $t_1 - t_n$ are variables or constants
2. An equality between 2 terms $(t_1 \approx t_2)$ or $(t_1 = t_2)$. $\approx$ means a custom predetermined equality like "Same blood type" for example

Formulas of $\mathcal{L}$

Formulas are built recursively from atoms by defining formation rules like connectives and quantifiers

1. Every atom in Atom($\mathcal{L}$) is a formula in Form($\mathcal{L}$)
2. If A is a formula so is $\neg A$
3. If A, B are formulas, $(A \vee B), (A \wedge B), (A \rightarrow B)$ and $(A \leftrightarrow B)$ are all formulas
4. If $A(u)$ is a formula where u is a free variable and x is a variable not occuring in $A(u)$, then $\forall x A(x)$ and $\exists x A(x)$ are formulas where $A(x)$ denotes the expression formed by replacing every occurrence of u with x

A formula follows a general recursive structure like this:



Closed vs Open formulas

A closed form is a formula with no free variables, meaning they are all in scope of a quantifier

$\forall x (A(x) \wedge B(x))$ → Closed
$\exists x (A(x) \rightarrow B(x)) \wedge A(x)$ → Open