This is the calculus one usually has in mind when using the word "logic". It is expressive enough for all of mathematics, except for those concepts that rely on a notion of construction or computation. However, dealing with more advanced concepts is often somewhat awkward and researchers often design specialized logics for that reason.

Our account of first-order logic will proceed similar to the one of propositional logic. We will present

- The *syntax*, or the formal language of first-order logic, that is symbols, formulas, sub-formulas, formation trees, substitution, etc.

- The *semantics* of first-order logic

- *Proof systems* for first-order logic, such as the axioms, rules, and proof strategies of the first-order tableau method and refinement logic

- The *meta-mathematics* of first-order logic, which established the relation between the semantics and a proof system

In many ways, the account of first-order logic is a straightforward extension of propositional logic. One must, however, be aware that there are subtle differences.

15.1   Syntax

Before Spring Break we introduced the syntax of first-order logic. Essentially it is an extension of propositional logic by quantification $\forall$ and $\exists$. Propositional variables are replaced by *n-ary predicate symbols* (P, Q, R) which may be instantiated with either *variables* (x, y, z, ...) or *parameters* (a, b, ...)

(1) *Atomic formulas* are expressions of the form $Pc_1..c_n$ where P is an n-ary predicate symbol and the c_i are variables or parameters.

(2) *Formulas* are built from atomic formulas using logical connectives and quantifiers.

  Every atomic formula is a formula.

  If A and B are formulas and x is a variable then (A), $\sim A$, $A \wedge B$, $A \vee B$, $A \supset B$, $(\forall x)A$, and $(\exists x)A$ are formulas.

(3) *Pure formulas* are formulas without parameters.

(4) The *degree $d(A)$* of a formula A is the number of logical connectives and quantifiers in A.

(5) The *scope* of a quantifier is the smallest formula that follows the quantifier.

  In $(\forall x)Px \vee Qx$ the scope of $(\forall x)$ is just Px, while Qx is outside the scope of the quantifier. To include Qx in the scope of $(\forall x)$ one has to add parentheses: $(\forall x)(Px \vee Qx)$.

  Note that the conventions about the scope of quantifiers differ in the literature.

(6) *Free and bound variables* are defined similarly to Second-Order Propositional Logic

A variable x occurs bound in A if it occurs in the scope of a quantifier. Any other occurrence of x in A is free.

(7) *Closed formulas* (or sentences) are formulas without free variables.

This is the default from now on.

(8) *Substitution*: $A|_a^x$ (or $A[a/x]$) is the result of replacing every free occurrence of the variable x in A by the parameter a.

The technical definition is similar to the one for P^2. However, since the term being substituted for x does not contain variables, capture cannot occur.

(9) *Subformulas* are defined similar to propositional logic.

The only modification is that for any parameter a the formula $A|_a^x$ is an immediate subformula of $(\forall x)A$ and $(\exists x)A$.

(10) The *formation tree* of a formula F is a representation of all subformulas of A in tree format.

That is, the root of the tree is F.

The sucessor of a formula of the form $\sim A$ is A.

The successors of $A \wedge B$, $A \vee B$, $A \supset B$ are A and B.

The successors of $(\forall x)A$ and $(\exists x)A$ are $A|_{a_i}^x$ for all parameters a_i. Note that quantifiers usually have infinitely many successors.

Atomic formulas have no successors.

15.2   Semantics

The semantics of first-order logic, like the one of propositional logic and P^2, is based on a concept of valuations. In propositional logic, it was sufficient to assign values to all propositional variables and then extend the evaluation from atoms to formulas in a canonical fashion. In P^2, the semantics of quantified formulas was defined in terms of the values of all immediate subformulas: $v[(\forall p)A] = (v|_f^p)[A] \wedge_\mathbb{B} (v|_t^p)[A]$.

In first-order logic, we will proceed in the same way. However, since we don't have propositional variables anymore, we have to explain the meaning of atomic formulas first.

The standard approach is to interpret parameters by elements of some *universe* $\mathcal{U}$ and n-ary predicates by subsets of $\mathcal{U}^n$. A closed formula $Pa_1..a_n$ then expresses the fact that the interpretations $k_i \in \mathcal{U}$ of the a_i, taken together as n-tuple $(k_1, .., k_n)$, form an element of the interpretation of P.

Smullyan's approach is similar to the above idea but avoids set theory altogether. Instead, he introduces *$\mathcal{U}$-formulas*, where the elements of the universe $\mathcal{U}$ are used as parameters and defines *first-order valuations* as canonical extensions of boolean valuations on the set $E^{\mathcal{U}}$ of all closed $\mathcal{U}$-formulas. The semantics of arbitrary formulas is then defined by a mapping φ from the set of parameters into $\mathcal{U}$.

Definition 1 *A **first-order valuation v of $E^{\mathcal{U}}$** is an assignment of truth values to elements of $E^{\mathcal{U}}$ such that*

(1) *v is a boolean valuation of $E^{\mathcal{U}}$, i.e.*

$v[\sim A] = t$ *iff* $v[A] = f$

$v[A \wedge B] = t$ *iff* $v[A] = t$ *and* $v[B] = t$

$v[A \vee B] = t$ *iff* $v[A] = t$ *or* $v[B] = t$

$v[A \supset B] = t$ *iff* $v[A] = f$ *or* $v[B] = t$

(2) *$v[(\forall x)A] = t$ iff $v[A|_k^x] = t$ for every $k \in \mathcal{U}$*

$v[(\exists x)A] = t$ iff $v[A|_k^x] = t$ for at least one $k \in \mathcal{U}$

All valuations can be defined as canonical extensions of *atomic valuations*, i.e. assignments of truth values to the atomic formulas in $E^{\mathcal{U}}$. A *valuation tree* for a formula A is the formation tree of A together with a consistent assignment of truth values to all the nodes in that tree.

Note that since formation trees are usually infinite, one cannot expect to *compute* the truth value of a formula A solely on the basis of a given atomic valuation.

As before, we can describe the semantics of formulas via truth sets and show that truth sets correspond to valuations in the sense that every first-order truth set is exactly the set of all formulas that are true under one (fixed) first-order valuation.

Definition 2 *A **first-order truth set S** (w.r.t. $\mathcal{U}$) is a subset of of $E^{\mathcal{U}}$ such that*

(1) *S statisfies the requirements on propositional truth sets, i.e.*

$A \in S$ *iff* $\sim A \notin S$

$A \wedge B \in S$ *iff* $A \in S$ *and* $B \in S$

$A \vee B \in S$ *iff* $A \in S$ *or* $B \in S$

$A \supset B \in S$ *iff* $A \notin S$ *or* $B \in S$

(2) *$(\forall x)A \in S$ iff $A|_k^x \in S$ for every $k \in \mathcal{U}$*

$(\exists x)A \in S$ iff $A|_k^x \in S$ for at least one $k \in \mathcal{U}$

The above definition can be extended to sentences with parameters as follows. Let φ be a mapping from the set of parameters to $\mathcal{U}$. For a formula A define A^φ to be the result of replacing every parameter a_i in A by $\varphi(a_i)$. We say that A is *true under φ and v* if $v[A^\varphi] = \mathsf{t}$.

The standard semantics of first-order formulas can be linked to the above as follows. Let E define the set of all closed formulas. An *interpretation of E* is a triple $I = (\mathcal{U}, \varphi, \iota)$, where $\mathcal{U}$ is an arbitrary set, φ is a mapping from the set of parameters to $\mathcal{U}$, and ι is a function that maps each n-ary predicate symbol P to a set $I(P) \subseteq \mathcal{U}^n$ (or an n-ary relation over $\mathcal{U}$).

An atomic sentence $Pa_1..a_n$ is *true under I* if $(\varphi(a_1), ..\varphi(a_n)) \in \iota(P)$. In this manner, every interpretation induces an atomic valuation v_0 (together with φ) and vice versa and from now on we will use whatever notion is more convenient.

A formula A is called *satisfiable* if it is true under at least one interpretation I (i.e. under at least one universe $\mathcal{U}$, one mapping φ, and one interpretation of the predicate symbols). I is also called a *model* of A. A is *valid* if A is true under every interpretation. These notions can be extended to sets of formula sin a canonical fashion.

It should be noted that there is a fine distinction between boolean valuations and first-order valuations. Boolean valuations can only analyze the propositional structure of formulas. They cannot evaluate quantified formulas and therefore have to treat them like propositional variables. In contrast to that first-order valuations can analyze the internals of quantified formulas and extract information that is unaccessible to boolean valuations.

For instance, a boolean valuations would interpret the logical structure of the formula $(\forall x)(Px \wedge Qx) \supset (\forall x)Px$ as $PQ \supset P$, which is obviously not a tautology. In contrast to that, every first-order valuation would go into the details of $(\forall x)(Px \wedge Qx)$ and $(\forall x)Px$ and evaluate to true. Thus the formula is valid, but not a tautology.

For the same reason, the formula $(\forall x)(Px \wedge Qx) \wedge (\exists x)(\sim Px)$ is truth-functionally satisfiable but not first-order satisfiable, since there is no first-order valuation (with a non-empty universe) that can make it true.

First-order valuations provide a more specific analysis than boolean valuations can give. They agree on quantifier-free formulas, however (Exercise!) and in that sense, first-order logic is a canonical extension of propositional logic.