First-Order Logic in Standard Notation – Basics

MATH VAULT

http://mathvault.ca

April 21, 2017

1 Vocabulary

Just as a natural language is formed with letters as its building blocks, the *First-Order Logic* (FOL) is formed with a series of primitive symbols consisting of:

- Constants — $a,\ b,\ c,\ d,\ a_1,\ b_1,\ c_1,\ d_1,\ a_2, \dots$
- Function Symbols — $f,\ g,\ h.\ f_1,\ g_1,\ h_1, \dots$

  Note: Each function symbol is equipped with an associated natural number (i.e., ≥ 1), called the *arity* of the functional symbol.
- Proposition Symbols — $A,\ B,\ C,\ A_1,\ B_1,\ C_1,\ \dots$
- Predicate Symbols — $=,\ P,\ Q,\ R,\ S,\ P_1,\ Q_1,\ R_1,\ S_1, \dots$

  Note: Similar to the function symbols, each predicate symbol has a built-in arity into it.
- Variables — $w,\ x,\ y,\ z,\ w_1,\ x_1,\ y_1,\ z_1, \dots$
- Connectives — $\neg,\ \wedge,\ \vee,\ \to,\ \leftrightarrow,\ \forall,\ \exists$
- Punctuation Symbols (within quotation marks) — "(", ")", ","

The set $\mathbf{V}$ of all these primitive symbols is then called the *vocabulary set* (of our rendition of FOL). A particular subset of $\mathbf{V}$ — the set of symbols from the first 4 categories (i.e., constants, function symbols, proposition symbols, predicate symbols), deserves a special attention. We called this resulting set (denoted by $\mathbf{L}$) the *lexicon*, with *lexical symbols* as its members. Much like lexical words in English, lexical symbols allow us to talk about entities and relations in the "real world."

Different kinds of symbols serve different purposes. Connectives are usually regarded as an essential feature of FOL. The puntuation symbols also play a decisive role in deambiguating expressions. Similarly, without variables, no universal (or existential) formula can be formed. And if only n variables are available, claims such as "there are at least n+1 objects" would not be expressible.

The lexical symbols, on the other hand, differ from the above symbols in an important way, as one can envision a FOL language without any constant, or without any propositional symbol, or without any function, or with only one predicate symbol. If anything, lexical symbols are characterized primarily by their optionality and lexicality.

2 String

A string is just a finite sequence of characters, which in turn belong to the vocabulary. Symbolically, a *string* is of the form $\mathbf{c_1}, \dots, \mathbf{c_n}$ (for some $n \in \mathbb{N}$), where each $\mathbf{c_i} \in \mathbf{V}$.

The fact that a string is finite and forms the basis of other higher concepts (e.g., term, formula, sentence) suggests that we are in effect disregarding any chunk that is infinitely long. To be fair, there are indeed first-order languages that accept terms and formulas of infinite length. However, these languages tend to be more interesting in theory than in practice, and require more heavy-duty set-theoretic machineries.

3 Terms

Much like proper nouns (e.g., Albert, Golden Bridge) and their derivatives (e.g., Albert's best friend, the width of Golden Bridge), terms in FOL are symbols created to "talk" about the individuals in our universe of discourse. A string is a *term* if and only if it satisfies one of the following conditions:

1. It is a constant, or a variable.
2. It is a functional term. That is, it is of the form $\mathbf{f}(\mathbf{t_1}, \dots, \mathbf{t_n})$ (for some $n \in \mathbb{N}$), where $\mathbf{f}$ is a function symbol of arity n and each $\mathbf{t_i}$ is a term.

The hallmark of terms is that after interpretation, each one of them will acquire a referent. In fact, it is for this purpose that they are introduced in the first place.

Since a term is a (finite) string, we can define the *degree* of a term to be the number of function symbols it contains. For example, if f is an unary function symbol, g a binary function symbol and h a ternary function symbol, then:

- $a,\ b,\ c,\ d_1$ are terms of degree 0.
- $f(b),\ g(a_1, b_3),\ h(c_2, d, a_1)$ are terms of degree 1.
- $f(f(d)),\ g(d_1, f(b)),\ h(d, c_6, g(c_3, b_1))$ are terms of degree 2.
- $f(g(a, f(b))),\ g(g(a, b_2), h(b, c, d)),\ h(f(c), g(b_5, a_6), d)$ are terms of degree 3.

It is a consequence of the definition that if a term has degree 0, then it is a constant or variable. Similarly, if a term has degree 1 or more, then it must be a functional term.

4 Formulas

Just as terms are tied to referents, formulas are tied to claims that one can make. Since terms are now defined, the (recursive) definition of a formula is now in order:

A (finite) string is a *formula* if and only if it satisfies one of the following conditions:

1. It is an atom. That is, either it is a proposition symbol, or of the form $\mathbf{P}(\mathbf{t_1},\ \dots,\ \mathbf{t_n})$, where $\mathbf{P}$ is a predicate symbol with arity n, and each $\mathbf{t_i}$ is a term.

   $\mathbf{Note}$: In the special case of equality claims of the form =$(\mathbf{t_1}, \mathbf{t_2})$, we adopt the convention of using infix notation instead (i.e., $\mathbf{t_1} = \mathbf{t_2}$), while acknowledging that in the strict sense, such an expression would be ungrammatical.
2. It is a negation (i.e., of the form $\neg\alpha$, where α is a formula).
3. It is a conjunction (i.e., of the form $(\alpha \wedge \beta)$, where α and β are formulas).
4. It is a disjunction (i.e., of the form $(\alpha \vee \beta)$, where α and β are formulas).
5. It is a conditional (i.e., of the form $(\alpha \to \beta)$, where α and β are formulas).
6. It is a biconditional (i.e., of the form $(\alpha \leftrightarrow \beta)$, where α and β are formulas).
7. It is an universal. That is, it is of the form $\forall\mathbf{x}\alpha$, where α is a formula and $\mathbf{x}$ a variable.

8. It is an existential. That is, it is of the form $\exists \mathbf{x}\alpha$, where α is a formula and $\mathbf{x}$ a variable.

Notice that parentheses are only needed when two formulas are combined with a binary connective (i.e., $\wedge, \vee, \rightarrow, \leftrightarrow$) (assuming that we don't count the parentheses found in terms and atoms). Also, since parentheses are always introduced in pairs, one can prove that in any formula, the number of left parentheses equals the number of right parentheses.

Similar to terms, formulas are also (finite) strings. This means that we can define the *degree* of a formula to be the number of connectives it contains. For example, if P is an unary predicate symbol, Q a binary predicate symbol and R a ternary predicate symbol, then:

- $A,\ B,\ P(x),\ Q(b,y),\ R(c,c,c),\ d=y$ are formulas of degree 0.
- $\neg a = x,\ (A \wedge P(b)),\ (Q(b,z) \vee R(a,w,c)),\ (P(b) \rightarrow P(b)),$ $(Q(x,y) \leftrightarrow Q(y,x)),\ \forall y P(a),\ \exists z R(a,y,x)$ are formulas of degree 1.
- $\neg(A \wedge P(a)),\ (Q(x,a) \vee \neg R(d,a,y)),\ (Q(a,b) \rightarrow (P(x) \wedge P(a))),$ $(R(x,y,z) \leftrightarrow \forall x Q(a,x)),\ \exists w \forall y Q(x,w)$ are formulas of degree 2.

It is a consequence of our definition that if a formula has degree 0, then it is an atom. Similarly, if a formula has degree 1 or more, then it must be a non-atomic formula (i.e., negation, conjunction, disjunction, conditional, biconditional, universal, existential).

5 Valuations — A Preliminary Look

We have been looking at how certain strings can arise from the primitive symbols. At this point, however, terms and formulas are simply certain sequences of symbols. In order to imbue them with meaning, we need *functions* that map the *lexical symbols* and *variables* to their semantic counterparts. Such a function is called a *valuation* (usually denoted by σ or τ). In addition to being a function, a valuation $\boldsymbol{\sigma}$ must satisfy all of the following assignment rules:

1. For each constant $\mathbf{c}$, it maps $\mathbf{c}$ to an individual in a predefined universe of discourse $\mathbf{U}$. This individual is then referred to as $\mathbf{c}^{\boldsymbol{\sigma}}$ (which reads "$\mathbf{c}$ under $\boldsymbol{\sigma}$").
2. For each variable $\mathbf{x}$, it also maps $\mathbf{x}$ to an individual in $\mathbf{U}$. This individual is then denoted by $\mathbf{x}^{\boldsymbol{\sigma}}$.
3. For each function symbol (with arity n) $\mathbf{f}$, it maps $\mathbf{f}$ to yet another function with domain $\mathbf{U^n}$ and range $\mathbf{U}$. This new function is then referred to as $\mathbf{f}^{\boldsymbol{\sigma}}$.

4. For each proposition symbol $\mathbf{A}$, it maps $\mathbf{A}$ to a truth value (i.e.,$\top$, $\bot$). This truth value is then denoted by $\mathbf{A}^{\boldsymbol{\sigma}}$.
5. For each predicate symbol $\mathbf{P}$ (with arity n), it maps $\mathbf{P}$ to a n-tuple relation in $\mathbf{U}^{\mathbf{n}}$. This relation is then called $\mathbf{P}^{\boldsymbol{\sigma}}$.

There is a type of valuation that deserves extra attention, for the role it plays in unmasking the truth values of existentials and universals. This kind of valuation is called a variant, as it differs from the usual valuation only in the assignment of one variable ($\mathbf{Note}$: We will make this claim more precise after a valuation is fully defined). More formally, $\boldsymbol{\sigma}(\mathbf{x}/\mathbf{u})$, a *variant* of $\boldsymbol{\sigma}$ on the variable $\mathbf{x}$ (where $\mathbf{u} \in \mathbf{U}$), is a valuation such that:

1. $\mathbf{x}^{\boldsymbol{\sigma}(\mathbf{x}/\mathbf{u})} = \mathbf{u}$
2. For each variable that is not $\mathbf{x}$ (here denoted by $\mathbf{y}$), $\mathbf{y}^{\boldsymbol{\sigma}(\mathbf{x}/\mathbf{u})} = \mathbf{y}^{\boldsymbol{\sigma}}$
3. For each *lexical symbol* $\mathbf{l}$, $\mathbf{l}^{\boldsymbol{\sigma}(\mathbf{x}/\mathbf{u})} = \mathbf{l}^{\boldsymbol{\sigma}}$

When there is no room for confusion, such a valuation can be referred to simply as a $\boldsymbol{\sigma}$-variant.

6 Valuations — The Full Version

The previous section tells us how a valuation interprets each lexical symbol and variable. However, that alone still won't allow us to determine the "meaning" of the terms and formulas. This is where the *Basic Semantic Definition* (BSD), originally attributed to Alfred Tarski , comes in.

In a nutshell, BSD is a set of rules that a valuation must satisfy. Once a valuation is equipped with BSD, it will allow us to (recursively) decrypt the referent of a term, or the truth value of a formula, by first decrypting the "meaning" of the components of the term (resp., formula).

We can now introduce the formal details of BSD, which essentially requires any valuation to satisfy the assignment rules for terms, and the assignment rules for formulas.

6.1 Assignment Rules for Terms

Given a valuation $\boldsymbol{\sigma}$, BSD requires that $\boldsymbol{\sigma}$ satisfy *all* the following assignment rules for terms:

1. If a term is a constant or a variable, then it is just a lexical symbol — The assignment rules from the previous section already told us how to interpret such a term.
2. If a term is of the form $\mathbf{f}(\mathbf{t_1},\ \dots,\ \mathbf{t_n})$ (where $\mathbf{f}$ is a n-ary function symbol, and each $\mathbf{t_i}$ is a term), then $[\mathbf{f}(\mathbf{t_1},\ \dots,\ \mathbf{t_n})]^{\boldsymbol{\sigma}} = \mathbf{f}^{\boldsymbol{\sigma}}(\mathbf{t_1}^{\boldsymbol{\sigma}},\ \dots,\ \mathbf{t_n}^{\boldsymbol{\sigma}})$.

6.2 Assignment Rules for Formulas

The assignment rules for terms allow us to interpreted every term unambiguously, but they constitute only a small portion of BSD. Given a valuation $\boldsymbol{\sigma}$, BSD also requires that $\boldsymbol{\sigma}$ satisfy *all* the following assignment rules for formulas:

1. If a formula is an atom, then it is either a proposition symbol, or of the form $\mathbf{P}(\mathbf{t_1},\dots,\ \mathbf{t_n})$. In the first scenario, the formula is just a lexical symbol, so the assignment rules from the previous section dealt with that already. In the second scenario,

$$[\mathbf{P}(\mathbf{t_1},\dots,\ \mathbf{t_n})]^{\boldsymbol{\sigma}} = \begin{cases} \top & \text{if } (\mathbf{t_1}^{\boldsymbol{\sigma}},\dots,\ \mathbf{t_n}^{\boldsymbol{\sigma}}) \in \mathbf{P}^{\boldsymbol{\sigma}} \\ \bot & \text{if } (\mathbf{t_1}^{\boldsymbol{\sigma}},\dots,\ \mathbf{t_n}^{\boldsymbol{\sigma}}) \notin \mathbf{P}^{\boldsymbol{\sigma}} \end{cases}$$

   In particular, $(\mathbf{t_1} = \mathbf{t_2})^{\boldsymbol{\sigma}} = \begin{cases} \top & \text{if } (\mathbf{t_1},\mathbf{t_2}) \in =^{\boldsymbol{\sigma}} \ \text{(i.e., } \mathbf{t_1} = \mathbf{t_2}) \\ \bot & \text{if } (\mathbf{t_1},\mathbf{t_2}) \notin =^{\boldsymbol{\sigma}} \ \text{(i.e., } \mathbf{t_1} \neq \mathbf{t_2}) \end{cases}$

2. If the formula is of the form $\neg\alpha$ (i.e., a negation), then $(\neg\alpha)^{\boldsymbol{\sigma}} = \begin{cases} \bot & \text{if } \alpha^{\boldsymbol{\sigma}} = \top \\ \top & \text{if } \alpha^{\boldsymbol{\sigma}} = \bot \end{cases}$
3. If the formula is of the form $(\alpha \wedge \beta)$ (i.e., a conjunction), then

$$(\alpha \wedge \beta)^{\boldsymbol{\sigma}} = \begin{cases} \top & \text{if } \alpha^{\boldsymbol{\sigma}} = \top \text{ and } \beta^{\boldsymbol{\sigma}} = \top \\ \bot & \text{otherwise} \end{cases}$$

4. If the formula is of the form $(\alpha \vee \beta)$ (i.e., a disjunction), then

$$(\alpha \vee \beta)^{\boldsymbol{\sigma}} = \begin{cases} \bot & \text{if } \alpha^{\boldsymbol{\sigma}} = \bot \text{ and } \beta^{\boldsymbol{\sigma}} = \bot \\ \top & \text{otherwise} \end{cases}$$

5. If the formula is of the form $(\alpha \to \beta)$ (i.e., a conditional), then

$$(\alpha \to \beta)^{\boldsymbol{\sigma}} = \begin{cases} \bot & \text{if } \alpha^{\boldsymbol{\sigma}} = \top \text{ and } \beta^{\boldsymbol{\sigma}} = \bot \\ \top & \text{otherwise} \end{cases}$$

6. If the formula is of the form $(\alpha \leftrightarrow \beta)$ (i.e., a biconditional), then

$$(\alpha \leftrightarrow \beta)^{\boldsymbol{\sigma}} = \begin{cases} \top & \text{if } \alpha^{\boldsymbol{\sigma}} = \beta^{\boldsymbol{\sigma}} \\ \bot & \text{otherwise} \end{cases}$$

7. If the formula is of the form $\forall \mathbf{x} \alpha$ (i.e., an universal), then

$$(\forall \mathbf{x} \alpha)^{\boldsymbol{\sigma}} = \begin{cases} \top & \text{if } \alpha^{\boldsymbol{\sigma}(\mathbf{x}/\mathbf{u})} = \top, \text{ for each } \mathbf{u} \in \mathbf{U} \\ \bot & \text{if } \alpha^{\boldsymbol{\sigma}(\mathbf{x}/\mathbf{u})} = \bot, \text{ for some } \mathbf{u} \in \mathbf{U} \end{cases}$$

8. If the formula is of the form $\exists \mathbf{x} \alpha$ (i.e., an existential), then

$$(\exists \mathbf{x} \alpha)^{\boldsymbol{\sigma}} = \begin{cases} \top & \text{if } \alpha^{\boldsymbol{\sigma}(\mathbf{x}/\mathbf{u})} = \top, \text{ for some } \mathbf{u} \in \mathbf{U} \\ \bot & \text{if } \alpha^{\boldsymbol{\sigma}(\mathbf{x}/\mathbf{u})} = \bot, \text{ for each } \mathbf{u} \in \mathbf{U} \end{cases}$$

7 Consequences of BSD

One immediate consequence of the Basic Semantic Definition is that each valuation can now interpret any term and any formula unambigrously (i.e., a term points to a unique referent, and a formula a unique truth value). Since variants are essentially valuations, this semantic unambiguity applies to all variants as well.

Note that it is generally false that $\boldsymbol{\sigma}(\mathbf{x}/\mathbf{u})$ agrees with $\boldsymbol{\sigma}$ on "everything" except $\mathbf{x}$ (for example, $f(y)^{\sigma}$ needs not be the same as $f(y)^{\sigma(y/154)}$). What we can claim, is that $\boldsymbol{\sigma}$ and $\boldsymbol{\sigma}(\mathbf{x}/\mathbf{u})$ agree on all lexical symbols and variables, except *perhaps* $\mathbf{x}$. However, it can be proved that for any term $\mathbf{t}$, if $\boldsymbol{\sigma}$ and $\boldsymbol{\sigma}(\mathbf{x}/\mathbf{u})$ agree on *all* the lexical symbols and free variables of $\mathbf{t}$ (a term only has free variables anyway), then $\mathbf{t}^{\boldsymbol{\sigma}} = \mathbf{t}^{\sigma(\mathbf{x}/\mathbf{u})}$. Similarly, given a formula α, if $\boldsymbol{\sigma}$ and $\boldsymbol{\sigma}(\mathbf{x}/\mathbf{u})$ agree on *all* the lexical symbols and *free* variables of α, then $\alpha^{\boldsymbol{\sigma}} = \alpha^{\boldsymbol{\sigma}(\mathbf{x}/\mathbf{u})}$.

In fact, these results are nothing but a special case of the so-called *Law of Semantic Agreement* of FOL, which states that given a term $\mathbf{t}$ (or a formula α), if $\boldsymbol{\sigma}$ and $\boldsymbol{\tau}$ agree on all the lexical symbols and *free* variables of $\mathbf{t}$ (resp., α), then $\mathbf{t}^{\boldsymbol{\sigma}} = \mathbf{t}^{\boldsymbol{\tau}}$ (resp.,

$\alpha^{\boldsymbol{\sigma}} = \alpha^{\boldsymbol{\tau}}$). Note that this theorem represents a major extension from its Propositional Logic counterpart, which states that given a propositional formula α, if $\boldsymbol{\sigma}$ and $\boldsymbol{\tau}$ agree on all the propositional symbols of α, then $\alpha^{\boldsymbol{\sigma}} = \alpha^{\boldsymbol{\tau}}$.

The Law of Semantic Agreement of FOL is a direct consequence of BSD, and has important implications in terms of establishing certain logical equivalences pertaining to null quantification, and to distributivity of quantifiers over boolean connectives (which in turn leads to the discovery of prenex forms).