First Order Logic

Declaration

This note was prepared based on the book *Logic for Computer Science* by A. Singha.

1 SYNTAX OF FL

The **alphabet of FL, *first order logic***, is the union of the following sets:

- $\{\top, \bot\}$, the set of **propositional constants**,
- $\{f_i^j : i, j \in \mathbb{N}\}$, the set of **function symbols**,
- $\{P_i^j : i, j \in \mathbb{N}\} \cup \{\approx\}$, the set of **predicates**,
- $\{x_0, x_1, x_2, \dots\}$, the set of **variables**,
- $\{\neg, \wedge, \vee, \rightarrow, \leftrightarrow\}$, the set of **connectives**,
- $\{\forall, \exists\}$, the set of **quantifiers**, and
- $\{), (, \}$, the set of **punctuation marks**.

Look at the subscripts and superscripts in the function symbols and predicates. The symbol f_i^j is a function symbol which could have been written as f_i; the superscript j says that the function symbol f_i^j has j arguments.

The superscript j in the function symbol f_i^j is referred to as its **arity**. The 0-ary function symbols do not require any variable or names to give us definite descriptions, that is, they themselves are definite descriptions. Thus the 0-ary function symbols are also termed as the **names**, or **individual constants**, or just **constants**.

The intention is to translate the definite descriptions that depend on j parameters. For example, the definite description 'author of Anna Karenina' will be written as $f_0^1(f_0^0)$, where f_0^0, a constant, stands for Anna Karenina; and f_1^1 stands for 'author of'. The superscript 1 in f_0^1 cannot be any other number, whereas the subscript could be different from 0.

Similarly, the superscripts in the predicates also refer to the arity of those. The relation 'brother of' is a binary predicate and is denoted by a predicate P_i^2; the subscript i may be one of $0, 1, 2, 3, 4, \ldots$. The 0-ary predicates do not have any gaps to be filled in so that they would become sentences; they are sentences already. Thus, 0-ary predicates P_i^0's are simply the *propositional variables* which, in some contexts, may not be analysed any deeper. This way, we really extend the syntax of PL to FL.

The symbol $\approx$ denotes the **equality or identity** predicate, assumed to be binary.

The symbol $\forall$ is called the **universal quantifier** and the symbol $\exists$ is called the **existential quantifier**.

Any string over the alphabet of FL is an **expression** (an FL-expression). The function symbols allow you to express very complex definite descriptions such as 'the left leg of mother of the eldest brother of father of the youngest sister of Gargi', by using composition of functions. All these definite descriptions, along with some others arising out of the use of variables are taken as **terms**. These are special types of expressions. The following is an inductive definition of terms.

Write t for a generic term. The grammar for **terms** is

$$t ::= x_i \mid f_i^0 \mid f_i^j(t, \ldots, t)$$

In the expression $f_i^j(t, \ldots, t)$, the symbol t occurs j times. The grammar says that the variables, and the constants or 0-ary function symbols are terms, and if $t_1, \ldots, t_j$ are terms with f_i^j, a j-ary function symbol, then the expression $f_i^j(t_1, \ldots, t_j)$ is a term.

A term is called **closed** iff no variable occurs in it.

Example 1.1. *f_5^0 is a term; it requires no arguments as its superscript is 0. Similarly, $f_0^1(f_0^0)$ and $f_5^1(f_3^2(f_0^0, f_1^0))$ are terms. Both of them are closed terms.*

f_5^1 is not a term; there is a vacant place as its superscript 1 shows. $f_5^1(f_1^0)$ is a closed term.

Similarly, $f_5^1(f_3^2(x_7, f_1^0))$ is also a term; it is not a closed term since a variable, namely, x_7 occurs in it.

Writing X for a generic formula, x for a generic variable, and t for a generic term, the grammar for **formulas** is:

$$\begin{aligned} X ::= \top \mid \bot \mid P_i^0 \mid (t \approx t) \mid P_i^m(t, \ldots, t) \mid \neg X \mid (X \wedge X) \mid (X \vee X) \\ \mid (X \to X) \mid (X \leftrightarrow X) \mid \forall x X \mid \exists x X \end{aligned}$$

In the expression $P_i^m(t, \ldots, t)$, the symbol t occurs m times. The grammar says that the special symbols $\top, \bot$, and the 0-ary predicates are (FL-) formulas. If P is any m-ary predicate, then for m terms $t_1, \ldots, t_m$, the expression $P(t_1, \ldots, t_m)$ is a formula. The equality predicate is written in infix notation; it allows $(s \approx t)$ as a formula for terms s and t.

The formulas might be obtained by using connectives as in PL or by prefixing a quantifier followed by a variable, to any other formula. Notice that all propositions of PL (generated from P_i^0's and connectives) are now treated as formulas (in FL).

The formulas in the forms $\top, \bot, P_i^0, (s \approx t)$, and $P_i^m(t_1, t_2, \ldots, t_m)$ are called **atomic formulas**; and other formulas are called **compound formulas**.

Example 1.2. *The following expressions are formulas:*

$$\top, \quad (\bot \to \top), \quad (f_0^1 \approx f_0^0), \quad (f_2^1(f_0^1, f_2^1(f_0^1)) \approx f_{11}^1), \quad P_2^1(f_1^1(x_5)),$$
$$\neg\forall x_3(P_2^5(f_1^1(x_5), f_0^1) \to P_3^0), \quad \forall x_2 \exists x_5(P_2^5(x_0, f_1^1(x_1)) \leftrightarrow P_3^1(x_1, x_5, x_6)).$$

Whereas the following expressions are not formulas:

$$\top(x_0), \quad P_1^1(f_0^1(f_0^0)), \quad f_0^1 \approx f_5^0,$$
$$(f_2^1(f_0^1, f_2^1(f_0^1)) \approx f_1^2(f_{12}^1, f_0^1)),$$
$$P_1^2(f_4^1(x_7), f_0^1), \quad \neg\forall x_1(P_2^5(f_0^1(x_2), x_3)),$$
$$\forall x_2 \exists x_5(P_2^5(x_0, f_0^1(x_1))),$$
$$\forall x_2 \exists x_5(P_2^5(f_0^1(x_1)) \leftrightarrow P_1^1(x_6)).$$

Theorem 1.1 (Unique Parsing). *Any formula X is in exactly one of the following forms:*

1. *$X = \top$.*
2. *$X = \bot$.*
3. *$X = P_i^0$ for a unique predicate P_i^0.*
4. *$X = (s \approx t)$ for unique terms s, t.*
5. *$X = P_i^j(t_1, \ldots, t_j)$ for unique predicate P_i^j and unique terms $t_1, \ldots, t_j$.*
6. *$X = \neg Y$ for a unique formula Y.*
7. *$X = (Y \circ Z)$ for a unique connective $\circ \in \{\wedge, \vee, \to, \leftrightarrow\}$, and unique formulas Y and Z.*
8. *$X = \forall x Y$ for a unique variable x and a unique formula Y.*
9. *$X = \exists x Y$ for a unique variable x and a unique formula Y.*

2 Scope and Binding

Let Y be a formula. A sub-string Z of Y is called a *subformula* of Y if and only if Z is a formula on its own. The *scope* of an occurrence of a quantifier occurring in Y is the subformula of Y starting with that occurrence (from that place to the right).

Example 2.1. *The subformulas of*

$$\forall x_1 \exists x_2 ((P_1^1(x_3) \wedge P_1^1(x_1)) \to P_1^1(x_2))$$

are:

$$\begin{aligned}
&P_1^1(x_3), \quad P_1^1(x_1), \quad P_1^1(x_2), \quad (P_1^1(x_3) \wedge P_1^1(x_1)),\\
&((P_1^1(x_3) \wedge P_1^1(x_1)) \to P_1^1(x_2)),\\
&\exists x_2((P_1^1(x_3) \wedge P_1^1(x_1)) \to P_1^1(x_2)),\\
&\forall x_1 \exists x_2((P_1^1(x_3) \wedge P_1^1(x_1)) \to P_1^1(x_2)).
\end{aligned}$$

The original formula has a single occurrence of $\forall$ *and a single occurrence of* $\exists$. *The scope of the occurrence of* $\forall$ *is the whole formula; and the scope of the occurrence of* $\exists$ *is*

$$\exists x_2((P_1^1(x_3) \wedge P_1^1(x_1)) \to P_1^1(x_2)).$$

An occurrence of a variable x in Y is a *bound occurrence* if and only if this occurrence is within the scope of an occurrence of $\forall x$ or $\exists x$ (a quantifier that uses it). If a variable occurs within the scopes of more than one occurrence of quantifiers using that variable, then this occurrence of the variable is said to be bound by the rightmost among all these occurrences of quantifiers.

An occurrence of a variable in a formula is called a *free occurrence* if and only if it is not bound. If there is a free occurrence of a variable x in a formula Y, we say that x occurs free in Y, and also, we say that x is a *free variable* of Y. A variable x is a *bound variable* of Y if and only if there is at least one bound occurrence of x in Y.

Example 2.2. *In the formula*

$$\exists x_2(P_2^1(x_2, x_1) \wedge \forall x_1 P_1^3(x_1)),$$

the scope of $\exists$ *is the whole formula; the scope of* $\forall$ *is the formula*

$$\forall x_1 P_1^3(x_1).$$

Here, all the occurrences of the variable x_2 *are bound; the first occurrence of* x_1 *is free while the last two occurrences of* x_1 *are bound occurrences. The variable* x_2 *is a bound variable of the formula while* x_1 *is both a free and a bound variable.*

A formula having no free variables is called a **closed formula** or a **sentence**. Thus, in a sentence, each occurrence of each variable is a bound occurrence. A formula that is not closed is called an **open formula**.

Example 2.3. *In the formula*

$$\forall x_1(\forall x_2(P_3^1(x_1, x_2, f_1^0) \wedge \forall x_1 P_2^1(x_1)) \to \exists x_3 P_3^1(x_3, x_1, x_3)),$$

all occurrences of x_1 are bound. All occurrences of x_1 are within the scope of the first $\forall$. The third and the fourth occurrences are also within the scope of the third $\forall$. However, these occurrences of x_1 are bound by the third occurrence of $\forall$, and not by the first $\forall$. The fifth occurrence of x_1 is bound by the first $\forall$.

In binding a variable, the scope of the innermost quantifier that uses the variable becomes relevant. The second occurrence of $\forall$ uses the variable x_2.

*All occurrences of variables x_1, x_2, x_3 are bound occurrences. The formula is a **sentence**.*

Convention 2.1. *We will drop the outer parentheses from formulas.*

Convention 2.2. *We will drop the superscripts from the predicates and function symbols; the arguments will show their arity. However, we must not use the same symbol with a different number of arguments in any particular context.*

For example, in the same formula, we must not use $P_5(x, y)$ and $P_5(f_2)$ since the first one says that P_5 is a binary predicate, and the second would require P_5 to be unary.

In the formula

$$\forall x_2 \exists x_5(P_4(x_0, f_1(x_1)) \leftrightarrow P_1(x_2, x_5, x_1)),$$

P_4 is a binary predicate, f_1 is a unary function symbol, and P_1 is a ternary predicate.

The following expression would not be considered a formula:

$$\forall x_2 \exists x_5(P_4(x_0, f_1(x_1)) \leftrightarrow P_4(x_2, x_5, x_1)).$$

Reason: *P_4 has been used once as a binary and once as a ternary predicate.*

Convention 2.3. *We will drop writing subscripts with variables, function symbols, and predicates, whenever possible. Instead, we will use $x, y, z, \ldots$ for variables, $f, g, h, \ldots$ for function symbols, and $P, Q, R, \ldots$ for predicates.*

The 0-ary predicates or propositional variables will be denoted by $A, B, C, \ldots$. Whenever we feel short of symbols, we may go back to writing them with subscripts.

Moreover, the 0-ary function symbols (which stand for names) will be written as $a, b, c, \ldots$, from the first few small letters of the Roman alphabet.

Following this convention, the formula

$$\forall x_2 \exists x_5(P_4(x_0, f_1(x_1)) \leftrightarrow P_1(x_2, x_5, x_1))$$

may be rewritten as

$$\forall x \exists y(P(z, f(u)) \leftrightarrow Q(x, y, u)).$$

Take care to see that each occurrence of x_2 has been replaced by x, each occurrence of P_4 has been replaced by P, etc.

Convention 2.4. *We will omit the parentheses and commas in writing the arguments of function symbols and predicates provided no confusion arises. If, however, some formula written this way is not easily readable, we will retain some of them.*

For example, the formula

$$\forall x \exists y (P(z, f(u)) \leftrightarrow Q(x, y, u))$$

may be rewritten as

$$\forall x \exists y (Pzf(u) \leftrightarrow Qxyu).$$

Similarly, the term $f(t_1, \ldots, t_n)$ will sometimes be written as $f(t_1 \ldots t_n)$ or as $ft_1 \ldots t_n$. And,

$$\forall x \exists y (Pzf(u) \leftrightarrow Qxyu)$$

may be rewritten as

$$\forall x \exists y (Pzfu \leftrightarrow Qxyu).$$

Convention 2.5. *We will have precedence rules for the connectives and quantifiers to reduce parentheses. We will preserve the precedence of connectives as in Propositional Logic (PL) and give the same precedence to the quantifiers and the connective $\neg$. That is,*

- $\neg, \forall x, \exists x$ *will have the highest precedence.* - $\wedge, \vee$ *will have the next precedence.* - $\rightarrow, \leftrightarrow$ *will have the lowest precedence.*

For example, the formula

$$\forall x_1 \neg (\exists x_2 ((P_1^1(f_1^2(f_0^0, f_0^1)) \wedge P_1^0) \rightarrow P_2^2(x_2, f_0^1) \leftrightarrow \forall x_3 ((P_1^1(f_0^5) \wedge P_2^1(x_1)) \rightarrow P_3^0))$$

is rewritten as

$$\forall x \neg (\exists y (Pf(a, b) \wedge A \rightarrow Qyb) \leftrightarrow \forall z (Pc \wedge Rx \rightarrow B)),$$

where the variables x_1, x_2, x_3 are rewritten as x, y, z; the function symbols $f_0^0, f_0^1, f_0^5, f_1^2$ are rewritten as a, b, c, f; and the predicates $P_1^1, P_1^0, P_2^2, P_2^1, P_3^0$ are rewritten as P, A, Q, R, B, respectively.

Caution: When considering the subformulas, scopes, and binding in a formula, we must take the formula in its original form, not in its abbreviated form.

For instance, $Px \rightarrow Py$ is not a subformula of

$$\forall x \exists y (Pz \wedge Px \rightarrow Py).$$

The given formula with adequate parentheses looks like

$$\forall x \exists y ((Pz \wedge Px) \rightarrow Py),$$

of which $Px \rightarrow Py$ is not a subformula.

The rewritten forms of formulas are, strictly speaking, not formulas. They are called **abbreviated formulas**. But we will regard them as formulas since they can be written back as formulas by following our conventions.

3 Substitutions

In what follows, we will require to substitute free variables by constants, other variables, or terms in general. We fix a notation for such substitutions and discuss what kind of substitutions are allowed.

We write $[x/t]$ for the substitution of the variable x by the term t, and then $(X)[x/t]$ for the formula obtained by replacing all free occurrences of the variable x with the term t in the formula X.

For example,

$$(Px \to Qx)[x/t] = Pt \to Qt$$

$$(\forall x Pxy)[x/t] = \forall x Pxy$$

$$\forall x(Pxy[x/t]) = \forall x Pty$$

$$\forall x \exists y((Px \wedge Qyx) \to Rzy)[y/t] = \forall x \exists y(Px \wedge Qyx) \to Rzt$$

$$(\forall x \exists y(Px \wedge Qyx) \to Rxy)[x/t] = \forall x \exists y(Px \wedge Qyx) \to Rty$$

We may abbreviate $(X)[x/t]$ to $X[x/t]$ if no confusion arises.

Note: In effecting a substitution, all and only free occurrences of a variable are replaced, and not every occurrence.

Consider the formula:

$$\forall x \exists y(Hx \to Fxy),$$

which you might have obtained by symbolizing the English sentence, *"Each human being has a father."*

To verify its truth, you are going to check whether for every object x, the formula $\exists y(Hx \to Fxy)$ is satisfied or not. Now, this 'every object x' can also stand for terms (definite descriptions, etc.).

For example, $f(a)$, the mother of a, is an object anyway. After replacement, the formula

$$(\exists y(Hx \to Fxy))[x/f(a)] = \exists y(Hf(a) \to Ff(a)y)$$

says that *"if the mother of a is a human being, then they have a father."*

Instead of $f(a)$, suppose you substitute $f(y)$, then you would obtain

$$(\exists y(Hx \to Fxy))[x/f(y)] = \exists y(Hf(y) \to Ff(y)y).$$

This formula says that *"there exists a human being such that if they have a mother, then the mother is their own father."* This is absurd, whereas the original sentence with our common understanding sounds plausible.

What went wrong?

Before the substitution, the occurrences of x were free in $\exists y(Hx \to Fxy)$. By substituting x as $f(y)$, all these new occurrences of y become bound. This is not a faithful substitution. Such a substitution is said to **capture the variable**. Our substitution should not capture a variable; it should not make a free occurrence bound.

Formal Definition

Let Z be a formula, and x, y be variables. The variable y is **free for** x in Z if and only if x does not occur free within the scope of any $\forall y$ or $\exists y$ in Z.

We observe that y is free for x in the formula Z is equivalent to the following condition: *After replacing each free occurrence of x by y in Z, the new occurrences of y remain free in the new formula.*

We say that a term t is **free for** the variable x in the formula Z if and only if each variable that occurs in t is free for x in the formula Z.

Free and Admissible Substitutions

Informally, it means that no variable of t is captured by replacing all free occurrences of x with t in Z. Thus, t is **free for** x means that t is allowed to be substituted in place of a free x, in some sense.

The formula $Z[x/t]$ is obtained from Z by replacing each free occurrence of the variable x by t, provided t is free for x in Z. The term $f[x/t]$ is obtained from $f(\cdot)$ by replacing each occurrence of x by t, in $f(\cdot)$.

Unless otherwise stated, whenever we use $Z[x/t]$, we assume that the term t is free for the variable x in the formula Z. Such substitutions are sometimes called **admissible substitutions**. In our notation, all substitutions are assumed to be admissible.

Example 3.1. *In the formula*

$$\exists y(Hx \to Fxy),$$

*y is **not** free for x since x occurs free within the scope of $\exists y$. This is the reason that after replacing this occurrence of x by $f(y)$, the new occurrence of y becomes bound; y is getting captured by the substitution.*

The variable x is free for x in $\exists y(Hx \to Fxy)$. Similarly, z is free for x in the formula $\exists y(Hx \to Fxy)$, as z does not occur in this formula at all.

*If t is a **closed term** (i.e., no variable occurs in it), then vacuously, each variable of t is free for each variable in each formula. Thus, t is free for each variable in each formula.*

If no variable of t is bound in a formula Y, then t is free for each variable occurring in Y. In the formula

$$\exists z(Pxy \leftrightarrow Qyz),$$

the term

$$t = f(x, y, f(a, b, c))$$

is free for each of the variables x, y, z. However, z is neither free for x nor free for y in this formula.

Only free occurrences of a variable can be substituted by a term, and that too, only when the substitution is admissible:

$$(\forall x Px)[x/t] = \forall x Px.$$

For replacing a variable by a term in a term, there is no such restriction:

$$f(x)[x/t] = f(t), \quad \textit{and} \quad f(x, y, z, g(a))[x/f(x)] = f(f(x), y, z, g(a)).$$