# CS112 Notes: First Order Logic: Part I

James Pustejovsky

September 6, 2004

## 1 Declarative Knowledge

In this handout we will examine a simple world, a syntactic description of it, and then look at the interpretive procedures for providing a "conceptualization" of the world within a semantics. We begin with a toy blocks world. Here is a sad picture of our world.

```
a
b   d
c   e
=====================
```

The universe of discourse, $\mathcal{D}$, consists of the following blocks, although they are not actually identified as any *sort* of entity.

$$\mathcal{D} = \{a, b, c, d, e\}$$

The "Directly on top of" function is an interrelationship between objects in the universe, defined as a partial function for this world, that we will call *doto*; this corresponds to the set of tuples below.

$$doto = \{< b, a >, < c, b >, < e, d >\}$$

The relation of "on top of" is a second kind of interrelationship among objects in the universe of discourse. We give *on* as the following set of tuples:

$$on = \{< a, b >, < b, c >, < d, e >\}$$

Notice that the set of tuples by itself does not uniquely define the kind of expression referred to by the object language. For the function *doto*, the maps one block into the block on top of it, if it exists. The evaluation of this function is an operation over the set of tuples stored for this function. The relation *on*, however, is evaluated for its truth-condition.

Now let's define an interpretation for all the elements of the language of first order logic into a semantics. An interpretation $I$ will be defined as a mapping between elements in the language of first order logic and of a conceptualization of those expressions. $I(\sigma)$ is the interpretation of the expression $\sigma$. $|I|$ will represent the universe of discourse.

For $I$ to be an interpretation, it must satisfy the following conditions:

(a) If $\sigma$ is an object constant, then $I(\sigma) \in |I|$.

(b) If $\pi$ is an $n$-ary function constant, then $I(\pi) : |I|^n \rightarrow |I|$.

(c) If $\rho$ is an $n$-ary relation constant, then $I(\rho) \subseteq |I|^n$.

With variables in expressions in the language, we will need an assignment function, which we will call $U$. A variable assignment is a function from the variables of a language to objects in the universe. We will put the assignment function together with the interpretation function into one function, which we'll call *term assignment*, $T_{IU}$.

(a) If $\tau$ is an object constant, then $T_{IU}(\tau) = I(\tau)$.

(b) If $\tau$ is a variable, then $T_{IU}(\tau) = U(\tau)$.

(c) If $\tau$ is a term of the form $\pi(\tau_1, \ldots, \tau_n)$ and $I(\pi) = g$ and $T_{IU}(\tau_i)\ x_i$, then $T_{IU}(\tau) = g(x_1, \ldots, x_n)$.

## 2 Satisfaction

An interpretation and variable assignment satisfy an equation if and only if the corresponding term maps into the same object. When this is the case, the two terms are said to be *coreferential*.

$\models_I (\sigma = \tau)[U]$ if and only if $T_{IU}(\sigma) = T_{IU}(\tau)$.

$I$ and $U$ satisfy an atomic sentence other than an equation if and only if the tuple formed from the objects designated by the terms in the sentence is an element of the relation designated by the relation constant.

$\models_I \rho(\tau_1, \ldots, \tau_n)[U]$ if and only if $< T_{IU}(\tau_1), \ldots, ; T_{IU}(\tau_n) >\in I(\rho)$.

Now we need to define satisfaction for all compositional constructions:

(a) $\models_I (\neg\phi)[U]$ if and only if $\neg \models_I \phi[U]$.

(b) $\models_I (\phi_1 \wedge \ldots \phi_n)[U]$ if and only if $\models_I (\phi_i)[U]$ for all $i = 1, \ldots, n$.

(c) $\models_I (\phi_1 \vee \ldots \phi_n)[U]$ if and only if $\models_I (\phi_i)[U]$ for some $i, 1 \leq i \leq n$.

(d) $\models_I (\phi \rightarrow \psi)[U]$ if and only if $\neg \models_I \phi[U]$ or $\models_I \psi[U]$.

(e) $\models_I (\phi \leftrightarrow \psi)[U]$ if and only if $\models_I (\phi \rightarrow \psi)[U]$ and $\models_I (\phi \leftarrow \psi)[U]$.

Universal and existential quantification in sentences are satisfied only if the sentence is satisfied for all assignments of the quantified variables.

(f) $\models_I (\forall\nu\phi)[U]$ if and only if for all $d \in |I|$ it is the case that $\models_I \phi[V]$, where $V(\nu) = d$ and $V(\mu) = U(\mu)$ for $\mu \neq \nu$.

(g) $\models_I (\exists\nu\phi)[U]$ if and only if for some $d \in |I|$ it is the case that $\models_I \phi[V]$, where $V(\nu) = d$ and $V(\mu) = U(\mu)$ for $\mu \neq \nu$.

# 3 Bound and Free Variables

In $\forall x F$ or $\exists x F$, we call $F$ the *scope* of the quantifier $\forall$ or $\exists$. An *occurrence* of a variable $x$ is said to be *bound* if it is inside the scope of a quantifier. Any other occurrence of a variable is called *free*.

Formulas without free variables are also called *closed formulas* or *sentential forms*. Formulas without variables are called *ground*.

$$\forall y \quad \overbrace{(\forall \overbrace{x \quad p(x)}^{scope} \quad \rightarrow \quad q(x,y))}^{scope}$$

The occurrence of $y$ is bound, as is the first occurrence of $x$. The second occurrence of $x$ is a free occurrence.

## 3.1 Substitutions

Substitution is a fundamental operation on terms and formulas that occurs in all inference systems for first-order logic. In the presence of quantification it is surprisingly complex.

By $F[x/s]$ we denote the result of substituting all free occurrences of $x$ in $F$ by the term $s$. Formally we define $F[x/s]$ by structural induction over the syntactic structure of $F$ as shown below.

$$
\begin{aligned}
x[x/s] &= s \\
x'[x/s] &= x' \;;\; \text{ if } x' \neq x \\
f(s_1, \ldots, s_n)[x/s] &= f(s_1[x/s], \ldots, s_n[x/s]) \\
\bot[x/s] &= \bot \\
\top[x/s] &= \top \\
p(s_1, \ldots, s_n)[x/s] &= p(s_1[x/s], \ldots, s_n[x/s]) \\
(u \approx v)[x/s] &= (u[x/s] \approx v[x/s]) \\
\neg F[x/s] &= \neg(F[x/s]) \\
(F \rho G)[x/s] &= (F[x/s] \rho G[x/s]) \;;\; \text{ for each binary connective } \rho \\
(QyF)[x/s] &= Qz((F[y/z])[x/s]) \;;\; \text{ with } z \text{ a ``fresh'' variable}
\end{aligned}
$$