Interpretations of PL (Model Theory)

1. Once again, observe that I’ve presented topics in a slightly different order from how I presented them in sentential logic. With sentential logic I discussed syntax and interpretations (or semantics) before I moved on to the game of tableau. This time I presented the rules for quantifier tableau before moving on to the formal semantics of predicate logic. As I mentioned above, doing things that way reinforces the idea that one can understand the notion of a formal proof (expressed by the single turnstile) independently of the notion of entailment (expressed by the double turnstile).

2. In order to hook up with our notion of entailment, we need interpretations to assign truth-values to sentences. That way, we can tell whether or not a given interpretation counts as a counterexample to a target sequent.

In PL, interpretations are called models, which are comprised of three things:

1. Interpretations begin by specifying a domain of discourse, \( \mathbb{D} \). Domains may be any non-empty sets of objects. (Funny, unpalatable things happen if you allow empty domains.)

2. Interpretations assign a denotation for each and every name (or individual constant) in a language: Denotations are specific objects from the domain. The denotation of some name \( n \) can simply be symbolized as \( /n/ \).

3. Finally, interpretations assign extensions to every predicate in the language. If \( P \) is an \( n \)-place predicate, then its extension will be some set of \( n \)-tuples (ordered pairs, triples, etc.) of objects form the domain. If \( P \) happens to be a 0-place predicate, then the interpretation directly assigns that letter a truth value, just as if it were an atomic formula in sentential logic.

Calculating Truth Values in PL:

Interpretations provide the basis upon which we can determine the truth or falsity of formulas in PL. In PL, we shall stipulate that only closed formulas (or sentences) receive a truth value.

1. We begin, as we did in SL, with atomic formulas. An atomic formula \( P^*n_1n_2…n_x \) is true just in case the \( n \)-tuple of objects denoted by \( n_1, n_2,…,n_x \) is in the extension of \( P \) (or if it is a 0-place predicate and has been directly assigned the value true).

Symbolically, 
\[
/P^*n_1n_2…n_x/ = T \text{ j.i.c. } \langle/n_1/,/n_2/,…,/n_x/> \text{ is in } /P^*/
\]

2. Truth-functional compounds are treated just as they were in SL.

3. We now turn to quantifier formulas. Here things get dicey.
3.1
The conceptual hurdle here is that unlike truth functional compounds, the truth values of quantifier formulas cannot be determined from the values of their constituents. For if a quantifier is non-vacuous, then some of its constituents will be open, and thus receive no truth value. Another challenge is presented by the fact that not every object in the domain of discourse needs to be denoted by some name.

3.2
We might be tempted to think we could simply say that an existential formula $\exists x \Phi$ is true just in case the extension of its scope $\Phi$ is non-empty, and that a universal formula is true just in case the extension of its scope includes every object in the domain. While this can work when the scope of a quantified formula is just a single predicate letter, it clearly will not work when the scope is itself a truth-functional compound. For we haven’t defined the extension of such compounds. What, for instance, would be the extensions of $(Fx & Gx)$ or $(Fx \rightarrow Gx)$?

3.3
The great early 20th Century logician Alfred Tarski solved this problem by first defining a notion of satisfaction of (open) truth-functional compounds by objects in the domain, and then he unpacked the truth of quantifier formulas in terms of such satisfaction. In effect, the set of objects satisfying a particular truth-functional compound played the role of that compound’s extension. An existential formula is true on an interpretation just in case there is some object in the domain that satisfies the scope of that formula, and a universal formula is true just in case its scope is satisfied by every object in the domain.

But there is another way to finesse these issues, which doesn’t require an appeal to an antecedently intelligible notion of satisfaction, which we would then have to define. Instead, it asks us to invent some means of referring to particular objects in the domain, and then defines the truth of quantifier formulas in terms of the truth of their instantiations. Officially, we shall pursue this sparser strategy. In the end, it will come to the same thing.

3.4
Here’s how the maneuver works. Imagine that we were to enrich our language with a new name, i (for ‘it’) and we enrich our interpretation by assigning ‘it’ a denotation.

A universal formula in our original language, $\forall \sigma \Phi$, is true just in case its instantiation $\Phi$ (‘it’ /$\sigma$) is true for every possible way to assign a denotation to ‘it’ from our domain of discourse, $\mathcal{D}$.

Similarly, an existential formula in our original language, $\exists \sigma \Phi$, is true just in case its instantiation $\Phi$ (‘it’ /$\sigma$) is true for some (at least one) way to assign a denotation to ‘it’.

3.5
Basically, these semantics tell us that an existential formula is true just in case there is some object in the domain, such that, if we had a name for it, then the instantiation of that name into the scope of the formula would come out true. And a universal formula will be true just in case every object in the domain is such that, if we had a name for it, then the instantiation of that name into the scope of the formula would be true.
3.6
A loose end: it turns out on these semantics that vacuous quantifiers will simply take on the same
truth values as their scope. That is, they are vacuous; they do nothing.

Simple enough, right??

From these rules for calculating truth values of quantifier formulas, you should be able to verify
that the basic quantifier equivalences hold: \(\forall x \Phi \equiv \neg \exists x \neg \Phi\) and \(\exists x \Phi \equiv \neg \forall x \neg \Phi\).

As a more complicated exercise, you might try to prove (by induction) that in a language
restricted to one place predicates, any formula is equivalent to one in which all universal
quantifiers range over conjunctions of literals (atomic formulas and their negations) and all
existential quantifiers range over disjunctions of literals. [A caveat: I haven’t thought this
completely through yet!!]

4. Natural Language Translations (Informal Interpretation): In thinking through some of the
exercises below, it is often useful to recast or to parse quantifier expressions into some natural
language analog. Here it is often useful to think of universal quantifiers operating over
conditionals as saying something to the effect that everything matching some condition
(expressed by the conditional’s antecedent) meets another condition (expressed by the
consequent). Similarly, it is useful to read the negations of existential formulas as denying the
existence of something with some (possibly complex or compound) characteristic. In natural
language, one typically pairs universal quantifiers with conditionals and existentials with
conjunctions. Although they are perfectly well-formed, an existential formula that operates over
a conditional is generally too weak to express much of anything that would be rendered
intelligible in natural language.

4.1 Here then are some quantifier expressions and their natural language analogs:

\(\forall x (Fx \rightarrow Gx) \rightarrow \text{All } F’s \text{ are } G’s. \text{ (Everything that is an } F \text{ is a } G.)\)

\(\neg \forall x (Fx \rightarrow Gx) \rightarrow \text{Not All } F’s \text{ are } G’s. \text{ (equivalent to Some } F’s \text{ are not } G’s.)\)

\(\forall x (Fx \& Px \rightarrow \neg Gx) \rightarrow \text{All } F’s \text{ that are } P \text{ are non-} G. \text{ (No } F \text{ that is a } P \text{ is a } G.)\)

\(\forall x (\exists y Fyx \rightarrow (Px \lor Qx)) \rightarrow \text{All } F’s \text{ that are } F’ed \text{ by something are either } P \text{ or } Q.\)

\(\neg \exists y (Py \& Ryy) \rightarrow \text{There aren’t any } P’s \text{ that } R \text{ themselves.}\)

\(\neg \exists y (Fy \& \forall x (Gx \rightarrow Hxy)) \rightarrow \text{There aren’t any } F’s \text{ that are } H’ed \text{ by every } G.\)

4.2 And here are some natural language translations of embedded quantifier constructions:

\(\exists y \forall x Rxy \rightarrow \text{Something is } R’ed \text{ by everything.}\)

\(\forall x \exists y Rxy \rightarrow \text{Everything } R’s \text{ something (Note that this is not equivalent to the sentence above!)}\)

\(\forall x \forall y (Rxy \rightarrow Ryx) \rightarrow \text{Anything that } R’s \text{ anything is } R’ed \text{ right back. (Symmetry)}\)

\(\forall x \forall y \forall z (Rxy \& Ryz \rightarrow Rxz) \rightarrow \text{Whenever something } R’s \text{ another, and that other } R’s \text{ yet something further, the first will also } R \text{ the third. (Transitivity)}\)
5. Some Exercises:

1. In a vein similar to that of 4.1 above, recast the following in (more or less) natural language:

   a. $\forall y (Py \rightarrow Fyy)$
   b. $\neg \forall y (Py \rightarrow (Qy \& \neg Ry))$
   c. $\forall y (Qy \rightarrow \exists z (Pz \& Byz))$
   d. $\exists x (Px \& \forall y Fyx)$
   e. $\exists x \exists y ((Px \& \neg Py) \& Bxy)$
   f. $(\forall y \exists x Bxy \rightarrow \forall y \exists x Byx)$
   g. $\forall x \forall y (Fxy \vee Fyx)$  [This is a property called connectivity.]
   h. $\forall x (\exists y Fxy \rightarrow \neg \exists z Fzx)$
   i. $\forall x \forall y (Fxy \rightarrow \forall z Fzx)$
   j. $\exists y (By \& \forall x ((Bx \& \neg Sxx) \rightarrow Syx))$  [This is a famous example; what follows from it?]

2. Construct simple models (or interpretations) that make all of the sentences in the following sets true. (In other words, show that the following sets are consistent.) It might help to sound these sentences out in natural language, in order to see just what it is that they “say.”

   a. $\exists x Fx, \exists x \neg Fx, \forall y (Fy \rightarrow Gy), \neg \forall y Gy$
   b. $\forall y (Py \rightarrow Qy), \exists x (Qx \& \neg Rx), \exists x (Px \& Rx)$
   c. $\forall y (Py \rightarrow Qy), \exists x (Qx \& \neg Rx), \neg \exists x (Px \& Rx)$
   d. $\forall y (Py \rightarrow Qy), \forall y (Py \rightarrow Ry), \neg \forall y (Qy \rightarrow Ry)$
   e. $\forall y (Py \rightarrow Qy), \exists x (Qx \& \neg Rx), \forall y (Py \rightarrow Ry)$
   f. $\forall y (Py \rightarrow Qy), \forall y (Py \rightarrow \neg Qy)$  [Think about what follows from the simultaneous truth of these two sentences.]
   g. $\exists x \exists y Rxy, \neg \exists y Ry y$
   h. $\exists x \exists y Rxy, \forall x \forall y (Rxy \rightarrow \neg Ryx)$
   i. $\exists x \exists y Rxy, \forall x \neg Rx, \forall x \exists y Rxy$
   j. $\forall xyz ((Rxy \& Ryz) \rightarrow \neg Rzx), \forall x \exists y Rxy$

   [Note that by showing the consistency of these sets, one thereby demonstrates that the negation of the last member of these sets cannot be entailed by the rest of the members.]
3. Now construct models demonstrating the consistency of the following sets. What’s the trick here?

a. \( \forall x \exists y Sxy, \forall x \sim Sxx, \forall x \forall y \forall z((Sxy \& Syz) \rightarrow Szx) \)

b. \( \forall x \exists y Rxy, \forall x \forall y \forall z((Rxy \& Ryz) \rightarrow Rzx), \forall x \forall y (Rxy \rightarrow \sim Ryx) \)

c. \( \forall x \exists y Fxy, \forall x \exists y Gxy, \forall x \sim Fxx, \forall x \sim Gxx, \forall x \forall y \forall z((Fxy \& Fyz) \rightarrow Fxz), \forall x \forall y (Fxy \rightarrow \sim Gxy) \)