Hamilton College
Russell Marcus
rmarcus1@hamilton.edu

Class 29 - November 3
Semantics for Predicate Logic

I. Proof Theory and Semantics

We have been working on the construction of formal theories.
A *theory* is just a set of sentences, which we call theorems.
A formal theory is a set of sentences of a formal language.
We identify a theory by its theorems, the set of sentences provable within that theory.
Some theories are finite.
Many interesting formal theories are infinite.
We construct infinite theories by using rules of inference which allow us to generate indefinitely many new theorems.

To construct a formal theory, we first specify a language, and its syntax: vocabulary and rules for well-formed formulas.
We have looked carefully at the syntax of both **PL** and **M**.
Once we have specified the wffs of a language, we can use that language in a theory.
The same language can be used in different theories.

To construct a theory, we specify the theorems.
We can list them.
Or, we can adopt some axioms and rules of inference.
For example, we could adopt the axioms of Euclidean geometry, or of Newtonian mechanics.
Such theories are usually placed within a background logical theory.
Their axioms are added to the logical axioms.
To construct formal physical theories, we generally add mathematical axioms as well.
Euclid and Newton were not careful about their rules of inference, or background logic in general.
Frege's logic, and the development of proof theory in the twentieth century, were responses to worries about the nature of inference, of logical consequence.
Frege wanted a gap-free logic, and so specified his rules of inference syntactically.
The meta-theoretic study of axioms and rules of inference is called proof theory.

Independent of proof theory, we can also provide a semantics for our language.
The study of the semantics of a formal language is called model theory.
In semantics, we assign truth values to the simple sentences of the language and truth conditions for the construction of complex sentences.
We can determine which wffs are logically true and which inferences are valid by using model theory.

At this point, the differences between proof theory and model theory may be obscure.
In proof theory, we specify the theorems and acceptable inferences.
In model theory, we characterize logical truth and validity.
In propositional logic, the theorems were exactly the logical truths.
So, proof theory and model theory have the same results for propositional logic.

A formal theory, like our system of propositional logic, is called complete when all the logically true wffs are provable.
A theory is called sound when every provable formula is logically true.
PL is both complete and sound.

In more sophisticated theories, proof separates from truth
Gödel's first incompleteness theorem shows that in theories with some minimal amount of mathematical strength, there will be true sentences that are not provable.
Gödel uses arithmetic to allow a formal theory to state properties like provability within the theory.
He constructs a predicate, 'is provable' that holds of sentences only with specific, statable arithmetic properties.
Then, he constructs a sentence that says, truly, of itself that it is not provable.
Since it is true, it is not provable.
Thus, in theories which allow the Gödel construction, model theory and proof theory provide different results.

In **PL**, our semantics mainly consisted of constructing the truth tables.
We simply interpret the sentences of **PL**, by assigning $\top$ or $\bot$ to each atomic sentence.
Then, we assign truth values to complex propositions by combining, according to the truth table definitions, the truth values of the atomic sentences.
Since we have only 26 simple terms, the capital English letters, there are only 2^{26} = ~6.7 million possible interpretations.
That is a large number, but it is a finite number.
A more useful language will have infinitely many simple terms: P, P', P'', P'''...
A language with infinitely many formulas will have an even greater infinitely many interpretations.
Still, since we are working with only two truth values, we can determine the logical truths even in a language with infinitely many variables.
We just look at the truth tables.

In **PL**, our proof system was our eighteen rules of natural deduction.
Systems of natural deduction seem to mirror ordinary reasoning.
The rules of inference are often intuitive.
Despite having no axioms, we were able to prove theorems using indirect and conditional methods.
Other proof systems use axioms.
Here is an example of an axiomatic system, I'll call PS in the language of propositional logic:

Formal system PS

Language and wffs: those of **PL**[1]

Axiom Schemata:

For any wffs α, β, and γ, statements of the following forms are axioms:

AS1: $\alpha \supset (\beta \supset \alpha)$

AS2: $(\alpha \supset (\beta \supset \gamma)) \supset ((\alpha \supset \beta) \supset (\alpha \supset \gamma))$

AS3: $(\sim\alpha \supset \sim\beta) \supset (\beta \supset \alpha)$

Rule of inference:

Modus ponens

---

[1] We do not need any of the wffs which use ∨, •, and ≡; see the lesson on adequacy.

PS and our system of natural deduction are provably equivalent, since they are equivalent languages, and both systems are complete.
Our system of natural deduction makes proofs shorter than they would be in axiomatic systems of logic.
Natural deduction systems have one main drawback: their metalogical proofs are more complicated.
When we reason about the system of logic we have chosen, we ordinarily choose an austere system.
If we want to show that a system of natural deduction is legitimate, we can show that it is equivalent to a more austere system.

Logical truth and validity were easy to define in **PL**, using the truth tables.
In **M**, and the other languages of predicate logic we will study, the semantics are more complicated.
Remember, separating the syntax of our language from its semantics allows us to treat our formal languages as completely uninterpreted.
We can take our proof system as an empty game of manipulating formal symbols.
Intuitively, we know what the logical operators mean.
But until we specify a formal interpretation, we are free to interpret them as we wish.
Similarly, our constants and predicates and quantifiers are, as far as the syntax of our language specifies, uninterpreted.
To look at the logical properties of the language, we interpret the logical particles variously, and see what is essential to the language itself, and what is imposed by an interpretation.

II. Interpretations

The first step in formal semantics is to show how to provide an interpretation of the language.
Then, we can determine the logical truths.
The logical truths will be the wffs that come out as true under every interpretation.

To define an interpretation in **M**, or in any of its extensions, we have to specify how to handle constants, predicates and quantifiers.
To interpret predicates and quantifiers, we use some set theory.
We need not add set theory to our object language, but we need it in our meta-language.
We interpret a first-order theory in four steps.

Step 1. Specify a set to serve as a domain of interpretation, or domain of quantification.
The domain of quantification will be the universe of the theory, the objects to which we are applying the theory.
We can consider small finite domains, like a universe of three objects: $U_1=\{1, 2, 3\}$; or $U_2=\{$Barack Obama, Hillary Clinton, and Rahm Emanuel$\}$.
Or, we can consider larger domains, like a universe of everything.

Technically, there is no set of everything; such a set would lead to paradox.
The set of woodchucks, for example, is not too large to be a set.
But, consider the set of things which are not woodchucks.
Among the things which are not woodchucks are sets.
If we take a set to be any collection, among the sets would be the set of all sets which are not members of themselves.
But that seemingly-well-defined set is paradoxical.
If it belongs to itself, then it can not belong to itself.

If it does not belong to itself, then it should.
This paradox, which Bertrand Russell found in Frege's set theory, shows that not every property, like the property of not being a woodchuck, determines a set.
In such cases, we can consider the collection a proper class instead of a set.
One must be careful handling proper classes, since they are explosive.
But, we will not run into difficulties with them here.

Step 2. Assign a member of the domain to each constant.
We introduced constants to be used as names of particular things.
In giving an interpretation of our language, we pick one thing out of the domain for each constant.
Different constants may correspond to the same object, just as an individual person or thing can have multiple names.

Step 3. Assign some set of objects in the domain to each predicate.
We interpret predicates as sets of objects in the domain of which that predicate holds.
If we use a predicate 'Ex' to stand for 'x has been elected president', then the interpretation of that predicate will be the set of things that were elected president.
In U_1, the interpretation of 'Ex' will be empty; in U_2 it will be {Barack Obama}.

Step 4. Use the customary truth tables for the interpretation of the connectives.
We are familiar with this part of the semantics from **PL**.

In order to determine the truth of sentences of our formal theory we first define satisfaction, and then truth for an interpretation.
Objects in the domain may satisfy predicates; ordered n-tuples may satisfy relations.
A wff will be satisfiable if there are objects in the domain of quantification which satisfy the predicates indicated in the wff.
A universally quantified sentence is satisfied if it is satisfied by all objects in the domain.
An existentially quantified sentence is satisfied if it is satisfied by some object in the domain.

A wff will be true for an interpretation if all objects in the domain of quantification satisfy the predicates indicated in the wff.

Let's take, for an example, the interpretation of a small set of sentences, with a small domain.

Sentences:
1. Pa • Pb
2. Wa • ~Wb
3. (∃x)Px
4. (x)Px
5. (x)(Wx ⊃ Px)
6. (x)(Px ⊃ Wx)

Domain: {Bob Simon, Rick Werner, Katheryn Doran, Todd Franklin, Marianne Janack, Russell Marcus, Martin Shuster}

a: Katheryn Doran
b: Bob Simon

Px: {Bob Simon, Rick Werner, Katheryn Doran, Todd Franklin, Marianne Janack, Russell Marcus, Martin Shuster}
Wx: {Katheryn Doran, Marianne Janack}

We can think of 'Px' as meaning that x is a professor of philosophy at Hamilton College.
We can think of 'Wx' as meaning that x is a woman professor of philosophy at Hamilton College.
But, the interpretation, speaking strictly, is strictly extensional, in terms of the members of the sets listed.

We call an interpretation on which all of a set of given statements come out true a *model*.
Given our interpretations of the predicates, not every sentence in our set is satisfied.
1-5 are satisfied.
But, 6 is not.
If we were to delete sentence 6 from our list, our interpretation would be a model.

To construct a model for a given set of sentences, we specify an interpretation, using the four steps above.
(Only the first three require any thought, here, since we will assume the standard truth-tables for the connectives.)

III. Exercise A. Construct a model for the following theory.

(x)(Px ⊃ Qx)
(∃x)(Px • Rx)
(∃x)(Px • ~Rx)
(∃x)(Qx • ~Rx)
Pa, Pb, Qc

IV. Logical Truth and Validity

A wff will be *logically true* if it is true for every interpretation.
For **PL**, the notion of logical truth was much simpler.
All we had to do was look at the truth tables.
For **M**, and even more so for **F**, the notion of logical truth is just naturally complicated by the fact that we are analyzing parts of propositions.
Here are a couple of logical truths of **M**:

LT1 (x)(Px ∨ ~Px)
LT2 Pa ∨ [(x)Px ⊃ Qa]

We can prove that LT1 and LT2 are logical truths by using our proof theory, or by using model-theoretic reasoning.
Let's do LT1 by indirect proof.

│1. ~(x)(Px ∨ ~Px)	AIP
│2. (∃x)~(Px ∨ ~Px)	1, CQ
│3. ~(Pa ∨ ~Pa)	2, EI
│4. ~Pa • ~~Pa	3, DM
5. (x)(Px ∨ ~Px)	1-4, IP, DN

QED

We can do LT2 by using a model-theoretic argument.

Suppose that 'Pa ∨ [(x)Px ⊃ Qa]' is not a logical truth.
Then there is an interpretation on which it is false.
On that interpretation, the object assigned to 'a' will not be in the set assigned to 'Px', and there is some counterexample to [(x)Px ⊃ Qa]
But, any counter-example to a conditional statement has to have a true antecedent.
So, every object in the domain of our supposed interpretation will have to be in the set assigned to 'Px'.
That contradicts the claim that the object assigned to 'a' will not be in the set assigned to 'Px'.
So, our assumption must be false: no interpretation will make that sentence false.
So, 'Pa ∨ [(x)Px ⊃ Qa]' is logically true.

QED

On Monday, we will discuss invalid arguments in predicate logic.
A valid argument is one which is valid under any interpretation, using any domain.
Our proof system has given us ways to show that an argument is valid.
But when we introduced our system of inference for **PL**, we already had a way of distinguishing the valid from the invalid arguments, using truth tables.
In **M**, we need a corresponding method for showing that an argument is invalid.
An invalid argument will have counter-examples, interpretations on which the premises come out true and the conclusion comes out false.

V. Solution to Exercise A

Step 1. Specify a set to serve as a domain of interpretation, or domain of quantification.
Domain = {Persons}
Step 2. Assign a member of the domain to each constant.
a = Barack Obama
b = Condoleezza Rice
c = Neytiri (from *Avatar*)
Step 3. Assign some set of objects in the domain to each predicate.
Px = {Human Beings}
Qx = {Persons}
Rx = {Males}
Step 4. Use the customary truth tables for the interpretation of the connectives.

Other interpretations are, of course, possible.