Philosophy 240: Symbolic Logic
Fall 2009
Mondays, Wednesdays, Fridays: 9am - 9:50am

Hamilton College
Russell Marcus
rmarcus1@hamilton.edu

Class 28 - November 2
Semantics for Predicate Logic (§8.5)

I. Proof theory and semantics

We have looked carefully at the syntax of both **PL** and **M**.
Once we have specified the wffs of a language, it is typical to do semantics (or model theory) and proofs.
In semantics, we assign truth conditions, and maybe truth values, to the sentences of the language.
In proof theory, we construct a system of inference using the formal language we have specified.
For **PL**, our semantics were the truth tables, and our proof theory was the system of natural deduction in Hurley's Chapter 7.
Other proof systems use axioms or trees.
Systems of natural deduction are preferable to trees, since they seem to mirror ordinary reasoning; the rules of inference are often intuitive.
Also, natural deduction systems make proofs shorter than they would be in axiomatic systems of logic.

Both semantics and proof theory are done in the meta-language, and the study of metalogic is mainly concerned with these two tasks.
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.
Here is an example of an austere axiomatic system, I'll call PS_R in the language of propositional logic:

Formal system PS_R
Language and wffs: those of **PL**[1]
Axioms:
For any wffs α, β, and γ
Axiom 1: $\alpha \supset (\beta \supset \alpha)$
Ax. 2: $(\alpha \supset (\beta \supset \gamma)) \supset ((\alpha \supset \beta) \supset (\alpha \supset \gamma))$
Ax. 3: $(\sim\alpha \supset \sim\beta) \supset (\beta \supset \alpha)$
Rule of inference:
Modus ponens

PS_R and Hurley's system of natural deduction are provably equivalent, since they are equivalent languages, and both complete.
Completeness, for the logician, means approximately that all the true wffs are provable.
Both systems are also sound, which means approximately that everything we can prove is also true.
Intuitively, we know what truth is.
But, we need to specify what we mean by 'true' for a formal system.
To do so, we engage in semantics, or model theory.
In model theory, we specify an interpretation of the language.
Casually, we know what the logical operators mean, but until we specify an interpretation, we are free to

[1] We do not need to use any of the wffs which use ∨, •, and ≡.

interpret them as we wish.
We can take our languages to be completely uninterpreted.
We can take our proof system as an empty game of manipulating formal symbols.

II. Interpretations

An interpretation of the language assign meanings to the various particles.
To specify an interpretation of the entire language, we also assign T or F to each atomic sentence of the language.
We assign truth values to complex propositions by combining, according to the truth table definitions, the truth values of the atomic sentences.

For **PL**, defining an interpretation is simple.
We only have 26 simple terms, the capital English letters.
Thus, there are only 2^{26} possible interpretations.
That is a large number, but it is a finite number.
A more useful language will have infinitely many simple terms.
A language with infinitely many formulas will have an even greater infinitely many interpretations.

To define an interpretation in **M**, or in any of its extensions, we have to specify how to handle predicates and quantifiers.
To interpret a first-order theory like **M**, we must use some set theory.
We need not add set theory to our formal language, just our metalanguage.
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, everything is too large to be a set; there is no set of everything since such a set would lead to paradox.
In such cases, we can take the domain to be what we call a proper class.

Step 2. Assign a member of the domain to each constant.

Step 3. Assign some set of objects in the domain to each predicate.
That is, we interpret predicates as sets of objects in the domain, sets 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}.
In Full First-Order Predicate Logic, we will assign ordered n-tuples to each relational predicate.
A two-place predicate is assigned an ordered pair, a three-place predicate is assigned a three-place relation, etc.
So, the relation 'Gxy', which could be understood as meaning 'is greater than' would be modeled in U_1 by {<2,1>, <3,1>, <3, 2>}

Step 4. Use the customary truth tables for the interpretation of the connectives.
Ordinarily, in order to determine the truth of sentenes 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 true iff there are objects or ordered n-tuples which satisfy it, that is if there are objects in the domain of quantification, which stand in the relations indicated in the wff.

We call an interpretation on which all of a set of given statements come out true a model.
A valid argument will have to be valid under any interpretation, using any domain.
An invalid argument may not have a counter-example in some domains.

We will not spend any further time on the semantics of predicate logic, except for proving argument invalid.
We will accept that our system of deduction is sound, so that any argument of which we can derive the conclusion is valid.
But, we need a method to show that an argument is invalid.

III. Invalidity In PL

Recall how we proved invalidity in propositional logic.
Consider an argument:

1. $A \supset B$
2. $\sim(B \cdot A)$ $\quad$ $/A \equiv B$

We lined up the premises and conclusion:

$A \supset B$ $\quad$ / $\quad$ $\sim(B \cdot A)$ $\quad$ // $\quad$ $A \equiv B$

We then assigned truth values to the component sentences to form a counterexample.
A counterexample is a valuation which makes the premises true and the conclusion false.

A	⊃	B	/	~	(B	·	A)	//	A	≡	B
F	**T**	T		**T**	T	F	F		F	**F**	T

So, the argument is shown invalid when A is false and B is true.

IV. The informal counter-example method

In predicate logic, we can use an informal method to prove an argument invalid.
Consider:

(x)(Wx ⊃ Hx)
(x)(Ex ⊃ Hx) / (x)(Wx ⊃ Ex)

We can provide an interpretation of the predicates that yields true premises but a false conclusion.

Wx: x is a whale
Ex: x is an elephant
Hx: x is heavy

So, 'all whales are heavy' and 'all elephants are heavy' are both true.
But, 'all whales are elephants' is false.

V. Exercises A. Show invalid, using the counterexample method:

1.
1. (x)(Ax ⊃ Bx)
2. Bj / Aj

2.
1. (∃x)(Ax · Bx)
2. Aa / Ba

3.
1. (x)(Hx ⊃ Ix)
2. (x)(Hx ⊃ ~Jx) / (x)(Ix ⊃ ~Jx)

VI. The method of finite universes

The informal counter-example method is fine for shorter, simpler arguments.
Some of you are probably smart enough to come up with something for an argument like:

1. (x)[Ux ⊃ (Tx ⊃ Wx)]
2. (x)[Tx ⊃ (Ux ⊃ ~Wx)]
3. (∃x)(Ux · Wx)
∴(∃x)(Ux · Tx)

But, it would be nice to have a method which requires less ingenuity.
If an argument is valid, then it is valid, no matter what we choose as our domain of interpretation.
Logical truths are true in all possible universes.
Even if our domain has only one member, or two or three, valid arguments should be valid.

Consider the following invalid argument:

(x)(Wx ⊃ Hx)
(x)(Ex ⊃ Hx) / (x)(Wx ⊃ Ex)

We will start by choosing a domain of one object in the universe.
We will call it 'a'.
Then:

(x)(Wx ⊃ Hx)	is equivalent to	Wa ⊃ Ha
(x)(Ex ⊃ Hx)	is equivalent to	Ea ⊃ Ha
∴(x)(Wx ⊃ Ex)	is equivalent to	Wa ⊃ Ea

Now, assign truth values, as in the propositional case to make the premises true and the conclusion false:

Wa	⊃	Ha	/	Ea	⊃	Ha	//	Wa	⊃	Ea
T	**T**	T		F	**T**	T		T	**F**	F

So, the argument is shown invalid in a one-member universe, where Wa is true, Ha is true, and Ea is false.

A specification of the assignments of truth values to the atomic sentences of the theory, as in the previous sentence, is called a counter-example.
Be careful not to confuse this use of 'counter-example' with the counter-example method.
When I ask you to specify counter-examples in the method of finite universes, I am asking for assignments of truth values of the atomic sentences, given your chosen domain.

The method of finite universes works for complex arguments, as well.
Consider the argument from the beginning of this section.

1. (x)[Ux ⊃ (Tx ⊃ Wx)]
2. (x)[Tx ⊃ (Ux ⊃ ~Wx)]
3. (∃x)(Ux · Wx)

∴(∃x)(Ux · Tx)

Ua	⊃	(Ta	⊃	Wa)	/	Ta	⊃	(Ua	⊃	~	Wa)	/	Ua	•	Wa	//	Ua	•	Ta
T	**T**	F	T	T		F	**T**	T	F	F	T		T	**T**	T		T	**F**	F

Counter-example: The argument is shown invalid in a one-member universe, where Ua is true; Ta is false; and Wa is true.

Not all invalid arguments are shown invalid in a one-member universe.
All we need is one universe in which they are shown invalid, to show that they are invalid.
Even if an argument has no counter-example in a one-member universe, it might still be invalid!

VII. Universes of more than one member

Consider the following invalid argument:

$(x)(Wx \supset Hx)$
$(\exists x)(Ex \cdot Hx)$ $\qquad$ $/ \ (x)(Wx \supset Ex)$

In a one-object universe, we have:

Wa	⊃	Ha	/	Ea	·	Ha	//	Wa	⊃	Ea
				F				T	**F**	F

There is no way to construct a counterexample, but the argument is invalid.
(I know, because I made it!)
We have to consider a larger universe.

If there are two objects in a universe, a and b:

$(x)\mathscr{F}x$	becomes	$\mathscr{F}a \cdot \mathscr{F}b$	because every object has $\mathscr{F}$
$(\exists x)\mathscr{F}x$	becomes	$\mathscr{F}a \lor \mathscr{F}b$	because only some objects have $\mathscr{F}$

If there are three objects in a universe, then

$(x)\mathscr{F}x$	becomes	$\mathscr{F}a \cdot \mathscr{F}b \cdot \mathscr{F}c$
$(\exists x)\mathscr{F}x$	becomes	$\mathscr{F}a \lor \mathscr{F}b \lor \mathscr{F}c$

Returning to the problem...
In a universe of two members, we represent the argument is equivalent to:

$(Wa \supset Ha) \cdot (Wb \supset Hb) \ / \qquad (Ea \cdot Ha) \lor (Eb \cdot Hb) \quad // \qquad (Wa \supset Ea) \cdot (Wb \supset Eb)$

Now, assign values to each of the terms to construct a counterexample.

(Wa	⊃	Ha)	·	(Wb	⊃	Hb)	/	(Ea	·	Ha)	∨	(Eb	·	Hb)
T	T	T	**T**	F	T	T		F	F	T	**T**	T	T	T

//	(Wa	⊃	Ea)	·	(Wb	⊃	Eb)
	T	F	F	**F**	F	T	T

The argument is shown invalid in a two-member universe, when

Wa: true	Wb: false
Ha: true	Hb: true
Ea: false	Eb: true

VIII. Constants

When expanding formulas into finite universes, constants get rendered as themselves.
That is, we don't expand a term with a constant when moving to a larger universe.
Consider:

(∃x)(Ax · Bx)
Ac /Bc

We can't show it invalid in a one-member universe.

Ac	·	Bc	/	Ac	//	Bc
	F	F				F

We must move to a two-member universe.
Here, we generate a counter-example.

(Ac	·	Bc)	∨	(Aa	·	Ba)	/	**Ac**	//	**Bc**
T	F	F	**T**	T	T	T		**T**		**F**

This argument is shown invalid in a two-member universe, when
Ac: true Bc: false
Aa: true Ba: true
Some arguments need three, four, or even infinite models to be shown invalid.

IX. Propositions whose main connective is not a quantifier

Consider the following argument:

(∃x)(Px · Qx)
(x)Px ⊃ (∃x)Rx
(x)(Rx ⊃ Qx) /(x)Qx

In a one-member universe, this argument gets rendered as:

Pa · Qa / Pa ⊃ Ra / Ra ⊃ Qa // Qa

But there is no counter-example in a one-member universe.

Pa	·	Qa	/	Pa	⊃	Ra	/	Ra	⊃	Qa	//	**Qa**
	F	F										**F**

In a two-member universe, note what happens to the second premise:

$$(Pa \cdot Qa) \vee (Pb \cdot Qb) \;/\; (Pa \cdot Pb) \supset (Ra \vee Rb) \;/\; (Ra \supset Qa) \cdot (Rb \supset Qb) \;//\; Qa \cdot Qb$$

Each quantifier is unpacked independently.
The main connective, the conditional, remains the main connective.
We can clearly see here the difference between instantiation and translation into a finite universe.
We can construct a counterexample for this argument in a two-member universe:

(Pa	$\cdot$	Qa)	$\vee$	(Pb	$\cdot$	Qb)	/	(Pa	$\cdot$	Pb)	$\supset$	(Ra	$\vee$	Rb)
	F	F	**T**	T	T	T				T	**T**	F	T	T

/	(Ra	$\supset$	Qa)	$\cdot$	(Rb	$\supset$	Qb)	//	Qa	$\cdot$	Qb
	F	T	F	**T**	T	T	T		F	F	T

This argument is shown invalid in a two-member universe, when
Pa: either true or false   Pb: true
Qa: false   Qb: true
Ra: false   Rb: true

(There is another solution. Can you construct it?)

X. Exercises B. Show each of the following arguments invalid by generating a counter-example using the method of finite universes.

1.
   1. $(x)(Ex \supset Fx)$
   2. $(\exists x)(Gx \cdot {\sim}Fx)$ / $(\exists x)(Ex \cdot {\sim}Gx)$

2.
   1. $(x)(Bx \supset {\sim}Dx)$
   2. ${\sim}Bj$ / Dj

3.
   1. $(x)(Hx \supset {\sim}Ix)$
   2. $(\exists x)(Jx \cdot {\sim}Ix)$ / $(x)(Hx \supset Jx)$

4.
   1. $(x)(Kx \supset {\sim}Lx)$
   2. $(\exists x)(Mx \cdot Lx)$ / $(x)(Kx \supset {\sim}Mx)$

5.
   1. $(\exists x)(Px \cdot Qx)$
   2. $(x)(Qx \supset {\sim}Rx)$
   3. Pa / $(x){\sim}Rx$

6.
   1. $(x)(Ax \supset Bx)$
   2. $(\exists x)(Dx \cdot Bx)$
   3. $(\exists x)(Dx \cdot {\sim}Bx)$ / $(x)(Ax \supset Dx)$

IX. Solutions

Sample answers to Exercises A
1. Ax: x is an apple; Bx: x is a fruit; j: a pear
2. Ax: x is a Met; Bx: x is a pitcher; a: Carlos Delgado
3. Hx: x is a desk; Ix: x has legs; Jx: x has arms

Sample answers to Exercises B
1. Shown invalid in a one-member universe, where Ga: true; Ea: false; Fa: false
2. Shown invalid in a one-member universe, where Bj: false; Dj: false
3. Shown invalid in a two-member universe, where Ha: true; Ia: false; Ja: false; Hb: true or false; Ib: false; Jb: true
4. Shown invalid in a two-member universe, where Ka: false; La: true; Ma: true; Kb: true; Lb: false; Mb: true.
5. Shown invalid in a two-member universe, where Pa: true; Qa: false; Ra: true; Pb: true; Qb: true; Rb: false
6. Shown invalid in a three-member universe, where Aa: true; Ba: true; Da: false; Ab: true or false; Bb: true; Db: true; Ac: false; Bc: false; Dc: true