Lecture 3: Logic and Set Theory

Elizabeth Coppock

Introduction to Pragmatics, Summer Semester 2012, HHU

> It is a commonplace of philosophical logic that there are, or appear to be, divergences in meaning between, on the one hand, at least some of what I shall call the formal devices—$\sim$, $\wedge$, $\vee$, $\supset$, $(\forall x)$, $(\exists x)$, (ιx) (when these are given a standard two-valued interpretation)—and, on the other, what are taken to be their analogues or counterparts in natural language—such expressions as *not, and, or, if, all, some* (or *at least one*), *the*. Some logicians may at some time have wanted to claim that there are in fact no such divergences; but such claims, if made at all, have been somewhat rashly made, and those suspected of making them have been subjected to some pretty rough handling.

1 Propositional Logic

Disjunction. Imagine you are playing a slot machine, and you win if you get three symbols of the same color, OR three symbols of the same shape (e.g. a red star, a red square, and a red triangle, or 2 red stars and a blue star). Of course, you also win if you get three symbols of the same color and the same shape. Let:

C = You get three of the same color.
S = You get three of the same shape.

Then we can represent "you get three of the same color or three of the same shape" as:

$$C \vee S$$

The $\vee$ symbol represents "or", a.k.a. *disjunction.* Under what circumstances is $C \vee S$ true?

- When C is true but S is false (e.g. blue diamond, blue diamond, blue triangle)
- When C is false but S is true (e.g. blue diamond, blue diamond, green diamond)
- When C and S are both true (e.g. 3 blue diamonds)

And no other circumstances (e.g. blue diamond, blue triangle, green diamond).

We can represent the meaning of $\vee$ using a *truth table:*

C	S	$C \vee S$
True	True	True
True	False	True
False	True	True
False	False	False

What we have represented here is called *inclusive* disjunction, because it includes the case where *both* of the so-called *disjuncts* are true.

Exclusive disjunction, where only one of the disjuncts can (and must) be true, is defined like this:

C	S	$C \oplus S$
True	True	**False**
True	False	True
False	True	True
False	False	False

This is closer to the intuitive meaning of *or* in "You can have chicken or beef" (but the inference to 'not both' is generally thought to be an implicature, while the conventional, semantic meaning of *or* is generally thought to be $\vee$).

Conjunction. Now suppose the game gets harder: you have to get all the same shape *and* all the same color in order to win. Then you would need for C *and* S to be true. This is represented:

$$C \wedge S$$

Truth table:

C	S	$C \wedge S$
True	True	True
True	False	False
False	True	False
False	False	False

Material implication. Let W = you win. You win if and only if you get all the same color or all the same shape. But it is also true that you win if you get all the same color. We can represent "If you get all the same color, then you win" as follows:

$$C \rightarrow W$$

Suppose there have been some questions about whether the slot machine is functioning correctly, and we want to check whether this claim is true. There are several cases to check:

- You get all the same color and you win. ✓
- You get all the same color and you don't win. **No!**
- You don't get all the same color but you still win (presumably you got all the same shape). ✓

- You don't get all the same color and you don't win (presumably you didn't get all the same shape). ✓

Thus the truth table for $C \to W$ is:

C	W	$C \to W$
True	True	True
True	False	False
False	True	True
False	False	True

Negation. The truth table for negation, written $\neg$, is simple. If C is true, then $\neg C$ is false. If C is false, then $\neg C$ is true:

C	$\neg C$
True	False
False	True

What is the negation of $C \wedge S$? To find out, we apply negation to $C \wedge S$:

C	S	$C \wedge S$	$\neg[C \wedge S]$
True	True	True	False
True	False	False	True
False	True	False	True
False	False	False	True

Note: We have to use the brackets [ ] to show that we are applying negation to the conjunction of C and S, rather than C.

Some equivalences. Fact: $P \to Q$ is equivalent to $\neg P \vee Q$.

P	Q	$\neg P$	$\neg P \vee Q$
True	True	False	
True	False	False	
False	True	True	
False	False	True	

Fact: $P \to Q$ is equivalent to $\neg Q \vee \neg P$:

P	Q	$\neg P$	$\neg Q$	$\neg Q \to \neg P$
True	True	False		
True	False	False		
False	True	True		
False	False	True		

Tautology: A logical expression that is true, no matter what values the variables it contains take on.

The expressions P and Q are variables of a certain kind – *propositional variables*. Propositional variables are variables whose value can be either true or false.

Each possible assignment of truth values to propositional variables is represented by a different *row of the truth table.*

So you can tell an expression is a tautology by looking at the pattern of Trues and Falses in the column underneath it in a truth table: If they're all true, then it is a tautology.

Here is a tautology: $P \vee \neg P$ (e.g. It is raining or it is not raining):

P	$\neg P$	$P \vee \neg P$
True	False	True
False	True	True

Contradiction. Two logical expressions are contradictory if for every assignment of values to their variables, their truth values are different. Example: P and $\neg P$.

P	$\neg P$
True	False
False	True

Another contradictory pair: $P \to Q$, and $P \wedge \neg Q$.

P	Q	$P \to Q$	$\neg Q$	$P \wedge \neg Q$
True	True	True	False	False
True	False	False	True	True
False	True	True	False	False
False	False	True	True	False

Syntax and semantics of PL. Propositional Logic is a *language,* with a syntax and a semantics.

- The **syntax** specifies the well-formed formulas of the language.
- The **semantics** specifies the semantic value of every well-formed formula.

Syntax. To generate all the well-formed formulas, we start with:

- an unbounded supply of propositional atoms $P, Q, R....P_1, P_2, ...Q_1, Q_2, ...$ etc.
- symbols: $\neg$, $\wedge$, $\vee$, $\to$, [,][1]

[1]The symbols vary. Some people use lowercase instead of uppercase letters for propositional atoms. A tilde ˜ is often used instead of $\neg$ for negation. Some logicians use $\supset$ instead of $\to$ (this is an older style, but it appears in the first sentence of the Grice paper). Some use an ampersand (&) instead of $\wedge$. Some use parentheses () instead of square brackets []. Use whatever is easiest for you.

And we use the following rules:

- Atoms: any propositional atom $p, q...$ is a wff
- $\neg$: if ϕ is a wff, then $[\neg\phi]$ is a wff
- $\wedge$: if ϕ and ψ are wffs, then $[\phi \wedge \psi]$ is a wff
- $\vee$: if ϕ and ψ are wffs, then $[\phi \vee \psi]$ is a wff
- $\rightarrow$: if ϕ and ψ are wffs, then $[\phi \rightarrow \psi]$ is a wff

Notes:

- Not every string using these symbols is a wff. For example: $[\neg][\wedge pq \rightarrow$ is meaningless.
- Parentheses/brackets can be omitted when the result is unambiguous.

Semantics. There are two possible semantic values for a formula:

- True (T) or 1
- False (F) or 0

(Sometimes people use 3 truth values, including an *undefined* value, or 4, adding an *over-defined* value.)

The semantics of a connective ($\rightarrow$, $\wedge$, $\vee$) describes the truth values of formulas given the truth values of the formulas it joins together. These are defined using truth tables as above.

The semantic value of a formula is always relative to an assignment of values to the propositional variables it contains. For example, suppose we have a function g that assigns the value 1 (True) to P and the value 0 (False) to Q.

$$g(P) = 1$$

$$g(Q) = 0$$

Let $[\![\phi]\!]^g$ signify the semantic value of formula ϕ under assignment g.

$$[\![P \rightarrow Q]\!]^g = \_\_\_\_$$

Consider another value assignment g' such that:

$$g'(P) = 1$$

$$g'(Q) = 1$$

What is the semantic value of $P \rightarrow Q$ under this value assignment?

$$[\![P \rightarrow Q]\!]^{g'} = \_\_\_\_$$

2 First Order Logic

In Propositional Logic, propositional atoms have no internal structure. In First Order Logic (or Predicate Logic, or Predicate Calculus), propositional atoms can be built up from smaller pieces.

$\textsc{love}(\textsc{John}, \textsc{Mary})$	'John loves Mary'
$\textsc{happy}(\textsc{John})$	'John is happy'
$\forall x[\textsc{happy}(x)]$	'Everyone is happy'
$\exists x[\textsc{even}(x) \wedge x > 1]$	'There are even numbers greater than 1'

In PL, the values of formulas are truth values (True or False). In FOL, there are other kinds of denotations as well, including:

- Individuals (like the number 2, Robin Möllemann)
- Sets (the set of even numbers)
- Relations (like the taller-than relation)
- Functions (like the mother-of function)

2.1 Sets

Set. An abstract collection of distinct objects which are called the *members* or *elements* of that set. Elements may be concrete (like the beige 1992 Toyota Corolla I sold in 2008, Robin Möllemann, or your computer) or abstract (like the number 2, the English phoneme /p/, or the set of all Swedish soccer players). The elements of a set are not ordered, and there may be infinitely many of them or none at all.

Empty set. The *empty set,* written $\varnothing$ or $\{\}$, is the set containing no elements.

Subset. A is a *subset* of B, written $A \subseteq B$, if and only if every member of A is a member of B.

$A \subseteq B$ iff for all x: if $x \in A$ then $x \in B$.

Proper subset. A is a *proper subset* of B, written $A \subset B$, if and only if A is a subset of B and A is not equal to B.

$A \subset B$ iff (i) for all x: if $x \in A$ then $x \in B$ and (ii) $A \neq B$.

Superset. A is a *superset* of B, written $A \supseteq B$, if and only if every member of B is ia member of A.

$A \supseteq B$ iff for all x: if $x \in B$ then $x \in A$.

Proper superset. A is a *proper superset* of B, written $A \supset B$, if and only if A is a superset of B and A is not equal to B.

$$A \supset B \text{ iff (i) for all } x\text{: if } x \in B \text{ then } x \in A \text{ and (ii) } A \neq B.$$

Powerset. The *powerset* of A, written $\wp(A)$, is the set of all subsets of A.

$$\wp(A) = \{S | S \subseteq A\}$$

Set Union. The *union* of A and B, written $A \cup B$, is the set of all entities x such that x is a member of A or x is a member of B.

$$A \cup B = \{x | x \in A \text{ or } x \in B\}$$

Set Intersection. The *intersection* of A and B, written $A \cap B$, is the set of all entities x such that x is a member of A and x is a member of B.

$$A \cap B = \{x | x \in A \text{ and } x \in B\}$$

Set Difference. The *difference* of A and B, written $A - B$ or $A \setminus B$, is the set of all entities x such that x is an element of A and x is not an element of B.

$$A - B = \{x | x \in A \text{ and } x \notin B\}$$

This is also known as the *relative complement* of A and B, or the result of *subtracting* B from A. $A - B$ can also be read, 'A *minus* B'.

Set complement. The *complement* of a set A, written A', is the set of all entities x such that x is not in A.

$$A - B = \{x | x \notin A\}$$

2.2 Exercises on sets

Domain: The Simpsons family.

Adults	=	{Marge, Homer}
Children	=	{Bart, Lisa}
Babies	=	{Maggie}
Pets	=	{Santa's Little Helper, Snowball}
Females	=	{Marge, Lisa, Maggie}
Males	=	{Homer, Bart}
Humans	=	{Marge, Homer, Bart, Lisa, Maggie}
Family	=	{Marge, Homer, Bart, Lisa, Maggie, Santa's Little Helper, Snowball}

Calculate the following:

1. True or false: $\varnothing \subseteq$Adults
2. True or false: Adults$\subseteq$Adults
3. $\wp$(Adults)
4. True or false: Children $\subset$ Humans
5. True or false: Humans $\subset$ Humans
6. True or false: Humans $\subseteq$ Humans
7. True or false: Humans $\subseteq$ Babies
8. Children $\cup$ Babies
9. {Bart, Lisa} $\cup$ {Maggie}
10. $\{x | x \in$ {Bart, Lisa} or $x \in$ {Maggie}}
11. True or false: Children $\cap$ Babies = Males $\cap$ Females
12. True or false: Humans $\subseteq$ {Adults,Humans}
13. True or false: Humans $\in$ {Adults,Humans}
14. How many elements are in the set {Adults,Humans}?

2.3 Ordered pairs and relations

Ordered pair. Sets are not ordered.

$$\{\text{Bart}, \text{Lisa}\} = \{\text{Lisa}, \text{Bart}\}$$

But the elements of an *ordered pair* written $\langle a, b\rangle$ are ordered. Here, a is the *first member* and b is the *second member*.

$$\langle \text{Bart}, \text{Lisa}\rangle \neq \langle \text{Lisa}, \text{Bart}\rangle$$

We can also have ordered triples.

$$\{\text{Bart}, \text{Lisa}, \text{Maggie}\} = \{\text{Maggie}, \text{Lisa}, \text{Bart}\}$$

$$\langle \text{Bart}, \text{Lisa}, \text{Maggie}\rangle \neq \langle \text{Maggie}, \text{Lisa}, \text{Bart}\rangle$$

Relation. A (binary) relation is a set of ordered pairs. For example, the 'older-than' relation among Simpsons kids:

$$\{\langle \text{Bart}, \text{Lisa}\rangle, \langle \text{Lisa}, \text{Maggie}\rangle, \langle \text{Bart}, \text{Maggie}\rangle\}$$

Note that this is a *set*. How many elements does it have?

Domain. The *domain* of a relation is the set of entities that are the first member of some ordered pair in the relation.

Range. The *range* of a relation is the set of entities that are the second member of some ordered pair in the relation.

Function. A function is a special kind of relation. A relation R from A to B is a function if and only if it meets both of the following conditions:

- Each element in the domain is paired with just one element in the range.
- The domain of R is equal to A

A function can be thought of as something that gives a single **output** for a given **input**.

Are these relations functions?

$$\{\langle \text{Bart}, \text{Lisa} \rangle, \langle \text{Lisa}, \text{Maggie} \rangle, \langle \text{Bart}, \text{Maggie} \rangle\}$$

$$\{\langle \text{Bart}, \text{Lisa} \rangle, \langle \text{Lisa}, \text{Maggie} \rangle\}$$

Notation: $F(a)$ denotes 'the result of applying function F to argument a' or F of a' or 'F applied to a'. If F is a function that contains the ordered pair $\langle a, b \rangle$, then:

$$F(a) = b$$

This means that given a as input, F gives b as output. Note: We saw this notation earlier with assignment functions (recall statements like $g(P) = 1$, etc.).

2.4 First-order logic: syntax and semantics

Recall that predicate calculus is a formal language; a logic. Logics have a syntax and a semantics.

Syntax: specifies which expressions of the logic are well-formed, and what their syntactic categories are.

Semantics: specifies which objects the expression correspond to, and what their semantic categories are.

Syntactic categories. Formulas are built up from the following expressions of the syntactic categories including: **(individual) constants**, **(individual) variables**, **predicate constants**, **logical connectives**, and **quantifiers**. Individual constants and individual variables are **terms**.

Semantic types. Each expression belongs to a certain semantic type. The types of FOL are: **individuals**, **sets**, **relations**, and **truth values** (True or False).

Expressions	Syntactic categories	Semantic Type
BART,MAGGIE	(individual) constant	individual
x	(individual) variable	individual
HAPPY, EVEN	unary predicate constant	set
LOVE, >	binary predicate constant	binary relation
LOVE(BART, MAGGIE)	(atomic) formula	truth value
HAPPY(x)	(atomic) formula	truth value
$x > 1$	(atomic) formula	truth value
$\forall x$[HAPPY(x)]	formula	truth value
$\exists x$[EVEN(x) $\wedge x > 1$]	formula	truth value

Note: **Prefix notation** puts the relation first; **infix** notation puts the relation in the middle. Prefix notation is proper but we can use infix notation for readability.

Composition rules. How to build formulas (metalanguage variables are Greek letters, except v):

- If π is a unary predicate and α is a term, then $\pi(\alpha)$ is a formula.
- If π is a binary predicate and α_1 and α_2 are terms, then $\pi(\alpha_1 \alpha_2)$ is a formula.
- If ϕ is a formula, then $\neg\phi$ is a formula.
- If ϕ is a formula and ψ is a formula, then $[\phi \wedge \psi]$ is a formula, and so are $[\phi \vee \psi]$, $[\phi \rightarrow \psi]$, and $[\phi \leftrightarrow \psi]$.
- If v is a variable and ϕ is a formula, then $\forall v \phi$ is a formula.
- If v is a variable and ϕ is a formula, then $\exists v \phi$ is a formula.

Semantics. Expressions of predicate calculus are *interpreted* in *models*. Models consist of a domain of individuals D and an interpretation function I which assigns values to all the constants:

$$\mathbf{M} = \langle D, I \rangle$$

Example. Reality model $M_r = \langle D, I_r \rangle$; fantasy model $M_f = \langle D, I_f \rangle$.

Domain for both models:

$$D = \{\text{Bart}, \text{Maggie}, \text{Lisa}\}$$

Interpretation functions:

I_r(BART) = I_f(BART) = Bart

I_r(MAGGIE) = I_f(MAGGIE) = Maggie

$I_r(\text{LOVE}) = \{\langle \text{Bart}, \text{Bart}\rangle, \langle \text{Maggie}, \text{Bart}\rangle\}$
$I_f(\text{LOVE}) = \{\langle \text{Bart}, \text{Bart}\rangle, \langle \text{Bart}, \text{Maggie}\rangle, \langle \text{Maggie}, \text{Maggie}\rangle, \langle \text{Maggie}, \text{Bart}\rangle\}$

$I_r(\text{HAPPY}) = \{\text{Bart}\}$
$I_f(\text{HAPPY}) = \{\text{Maggie}, \text{Bart}, \text{Lisa}\}$

Interpretation rules:

- **Constants**
  If α is a constant, then $[\![\alpha]\!]^M = \text{I}(\alpha)$.
- **Atomic formulae**
  If π is an n-ary predicate and $\alpha_1, ...\alpha_n$ are terms, then $[\![\pi(\alpha_1, ..., \alpha_n)]\!]^M = 1$ iff
  $$\langle [\![\alpha_1]\!]^M, ..., [\![\alpha_n]\!]^M \rangle \in [\![\pi]\!]^M$$
  If π is a unary predicate and α is a term, then $[\![\pi(\alpha)]\!]^M = 1$ iff $[\![\alpha]\!]^M \in [\![\pi]\!]^M$.
- **Negation**
  $[\![\neg\phi]\!]^M = 1$ if $[\![\phi]\!]^M = 0$; otherwise $[\![\neg\phi]\!]^M = 0$.
- **Connectives**
  $[\![\phi \wedge \psi]\!]^M = 1$ if $[\![\phi]\!]^M = 1$ and $[\![\psi]\!]^M = 1$; 0 otherise. Similarly for $[\![\phi \vee \psi]\!]^M$, $[\![\phi \rightarrow \psi]\!]^M$, and $[\![\phi \leftrightarrow \psi]\!]^M$.

Because HAPPY is a unary predicate and MAGGIE is a term,

$[\![\text{HAPPY}(\text{MAGGIE})]\!]^{M_f} = 1$ iff $[\![\text{MAGGIE}]\!]^{M_f} \in [\![\text{HAPPY}]\!]^{M_f}$, i.e. iff Maggie $\in \{\text{Maggie}, \text{Bart}, \text{Lisa}\}$.
$[\![\text{HAPPY}(\text{MAGGIE})]\!]^{M_r} = 1$ iff $[\![\text{MAGGIE}]\!]^{M_r} \in [\![\text{HAPPY}]\!]^{M_r}$, i.e. iff Maggie $\in \{\text{Bart}\}$.

But what about variables? How do we interpret a formulas like this:

$\forall x\ \text{HAPPY}(x)$	'for all x, x is happy'
$\exists x\ \neg\text{HAPPY}(x)$	'there exists an x such that it is not the case that x is happy'
$\text{HAPPY}(x)$	'x is happy'

In order to interpret formulas with variables, we need to make interpretation relative to a model *and an assignment function*:

$$[\![\phi]\!]^{M,g}$$

An assignment function assigns individuals to variables. Example:

$$g_1 = \begin{bmatrix} x & \rightarrow & \text{Maggie} \\ y & \rightarrow & \text{Bart} \\ z & \rightarrow & \text{Bart} \end{bmatrix}$$

Informally, $\forall x\ \text{HAPPY}(x)$ is true iff: no matter which individual we assign to x, $\text{HAPPY}(x)$ is true. In other words, for all elements in the domain d, $d \in [\![\text{HAPPY}]\!]$.

Informally, $\exists x\ \text{HAPPY}(x)$ is true iff: we can find some individual to assign to x such that $\text{HAPPY}(x)$ is true. In other words, there is some element in the domain d such that $d \in [\![\text{HAPPY}]\!]$.

The assignment function determines what x is assigned to. Formally: $[\![x]\!]^{M,g} = g(x)$. Let's interpret $\text{HAPPY}(x)$ using the reality model M_r and the assignment function g_1.

$[\![\text{HAPPY}(x)]\!]^{M_r,g_1} = 1$ if and only if:
$[\![x]\!]^{M_r,g_1} \in [\![\text{HAPPY}]\!]^{M_r,g_1}$, which is true if and only if:
$g_1(x) \in I_r(\text{HAPPY})$, which is true if and only if:
Maggie $\in \{\text{Bart}\}$, which is false. So $[\![\text{HAPPY}(x)]\!]^{M_r,g_1} = 0$.

But if we use g_2, where x is mapped to Bart, then the value is 1. $[\![\text{HAPPY}(x)]\!]^{M_r,g_2} = 1$.

$$g_2 = \begin{bmatrix} x & \rightarrow & \text{Bart} \\ y & \rightarrow & \text{Bart} \\ z & \rightarrow & \text{Bart} \end{bmatrix}$$

New interpretation rules:

- **Constants**
  If α is a constant, then $[\![\alpha]\!]^{M,g} = \text{I}(\alpha)$.
- **Variables – all new!**
  If α is a variable, then $[\![\alpha]\!]^{M,g} = g(\alpha)$.
- **Atomic formulae**
  If π is an n-ary predicate and $\alpha_1, ...\alpha_n$ are terms, then $[\![\pi(\alpha_1, ..., \alpha_n)]\!]^{M,g} = 1$ iff
  $$\langle [\![\alpha_1]\!]^{M,g}, ..., [\![\alpha_n]\!]^{M,g} \rangle \in [\![\pi]\!]^{M,g}$$
  If π is a unary predicate and α is a term, then $[\![\pi(\alpha)]\!]^{M,g} = 1$ iff $[\![\alpha]\!]^{M,g} \in [\![\pi]\!]^{M,g}$.
- **Negation**
  $[\![\neg\phi]\!]^{M,g} = 1$ if $[\![\phi]\!]^{M,g} = 0$; otherwise $[\![\neg\phi]\!]^{M,g} = 0$.
- **Connectives**
  $[\![\phi \wedge \psi]\!]^{M,g} = 1$ if $[\![\phi]\!]^{M,g} = 1$ and $[\![\psi]\!]^{M,g} = 1$; 0 otherise. Similarly for $[\![\phi \vee \psi]\!]^{M,g}$, $[\![\phi \rightarrow \psi]\!]^{M,g}$, and $[\![\phi \leftrightarrow \psi]\!]^{M,g}$.
- **Universal quantification – all new!**
  $[\![\forall v\phi]\!]^{M,g} = 1$ iff for all $d \in D$, $[\![\phi]\!]^{M,g'} = 1$, where g' is an assignment function exactly like g except that $g'(v) = d$.
- **Existential quantification – all new!**
  $[\![\exists v\phi]\!]^{M,g} = 1$ iff there is a $d \in D$ such that $[\![\phi]\!]^{M,g'}$, where g' is an assignment function exactly like g except that $g'(v) = d$.