Lecture 5

Predicate Logic 1
Predicate Logic

• No significant advances in logical theory were possible until Aristotle’s (384 BC -322 BC) conception of logical form was replaced by one that was more adequate for the complexities of propositional structure.

• 1879: Frege, in his famous *Begriffsschrift* (a short book on logic) introduced a new notion of logical form, and an elaborate, rigorously defined notation for representing it.

• Frege’s formal system is known as *predicate logic*, or as the *predicate calculus*. 
Predicate Logic

• Predicate logic is concerned with logical relations that hold within a sentence.
• In predicate logic a formula can be composed of a number of individual term(s) and a predicate. For example, a sentence like

\[ \text{Pluto is a dog} \]

• corresponds to a \textbf{proposition} which is represented as

\[ D(p) \]

• This proposition contains
  – a \textbf{one-place predicate} \( D \), which stands for the property of being a dog,
  – the \textbf{term} \( p \), which stands for the individual Pluto. The proposition is true just in case the individual Pluto is a member of the set of individuals that are dogs.
Predicate Logic

• Propositional logic deals with logical relations that hold between sentences.

• Predicate logic contains propositional logic as one of its parts.

Pluto is a dog and Dean is a dog.

\[ D(p) \land D(d) \]
Predicate Logic

- **Syntax of predicate logic**
  
  The **VOCABULARY** consists of:

1. **INDIVIDUAL TERMS**
   
   a. individual constants: j, m, ...
   
   b. individual variables: x, y, z, ... (also subscripted: \(x_1\), \(x_2\), ..., \(x_n\))

2. **PREDICATE TERMS**
   
   a. predicate constants: P, Q, R, ... (capital letters)
   
   b. predicate variables: \(\Phi\) (“phi”, capital Greek letters)

3. **CONNECTIVES**
   
   five connectives of propositional logic: \(\neg\), \(\land\), \(\lor\), \(\rightarrow\), \(\leftrightarrow\)

4. **QUANTIFIERS:**
   
   \(\forall\) universal quantifier
   
   \(\exists\) existential quantifier

5. **AUXILIARY SYMBOLS:** ( ) and [ ]
Predicate Logic

• Syntax of predicate logic

The SYNTACTIC RULES generate (i.e., recursively define) the set of well-formed formulas (wff’s) of the language of predicate logic. The set of propositions is a proper subset of this set.

• Basic case
• Recursive rules
• Exclusion clause
Predicate Logic

• Syntax of predicate logic

• Basic case
  If $\alpha$ is a basic expression of a given category, then $\alpha$ is a well-formed expression of that category.
Predicate Logic

• Syntax of predicate logic

Recursive rules

1. If \( P \) is an \( n \)-place (i.e., \( n \)-ary) predicate constant and \( t_1, \ldots, t_n \) are individual terms (constants or variables), then \( P(t_1, \ldots, t_n) \) is a well-formed formula \( \phi \).

Note: the metavariables \( \phi \) (Phi), \( \psi \) (Psi) (lower case Greek letters), etc. are used to stand for well-formed formulas.

2. If \( \phi \) and \( \psi \) are well-formed formulas, then
   \[ \neg \phi, [\phi \land \psi], [\phi \lor \psi], [\phi \rightarrow \psi], [\phi \leftrightarrow \psi] \] are well-formed formulas.

3. If \( \phi \) is a well-formed formula and \( x \) is an individual variable, then
   \[ \forall x \phi \] (universal generalization of \( \phi \)) and
   \[ \exists x \phi \] (existential generalization of \( \phi \)) are well-formed formulas.
Predicate Logic

• Syntax of predicate logic

**Exclusion clause**
Nothing else is a formula of predicate logic.
The formulas of the language of predicate logic can only be generated by finite numbers of applications of recursive rules 1.-3.
Predicate Logic

- This system is called the first-order predicate logic, FOL.
- It is a ‘first-order’ logic, because predicates and quantifiers always range over individuals, and not over higher-order entities like properties and relations.
Predicate Logic

The **VOCABULARY** can be divided into two main types of symbols

1. **Variables**
2. **Constants**

There are two main types of **constants**:

1. **Non-logical**
   A non-logical constant (an individual constant like \( p \) standing for the dog named Pluto, a predicate constant like \( D \) standing for the property of being a dog) only has meaning or semantic content when one is assigned to it by means of an interpretation. Consequently, a sentence containing a non-logical constant lacks meaning except under an interpretation, so a sentence is said to be true or false under an interpretation.

2. **Logical**
   The interpretation of logical constants is held fixed
   \( \neg, \land, \lor, \rightarrow, \leftrightarrow, \forall, \exists, ( ) \) and [ ]
Predicate Logic

• **Two kinds of individual terms**
  • individual constants and
  • individual variables.

• Individual constants refer to or denote specific individuals.
• The simplest examples of individual constants are **proper names** (or rigid designators) like *John*.
  Each proper name is taken to be connected to exactly one individual in the universe of discourse in the model.
Predicate Logic

• Proper names

• In the universe of discourse of semantic models, each proper name corresponds to exactly one individual. This is clearly different from the way in which proper names work in daily life. Of course, there is more than just one person whose name is John. If a given name applies to more than one person, we always have the option of uniquely identifying people with the same name by assigning them different social security numbers, for example.

• We connect each proper name with exactly one individual in the model, because we need to ensure that proper names refer uniquely, which in turn is required for the evaluation of the truth value of sentences.

• In model-theoretic semantics, we assume that proper names are rigid designators, and have the same unique denotation or reference (the same particular object or fact) in all the possible worlds and times. This was proposed by the philosopher Saul Kripke (1972), who also takes natural kind terms like water to be rigid designators, apart from proper names. All other terms are non-rigid designators.
Predicate Logic

• **Definite Descriptions**

• Other types of expressions that may refer to a particular individual in the universe of discourse of semantic models, are *definite descriptions* like

  *the current president of Brazil*
  *the tallest man in the world*
  *the bag lying on this desk*
  *my mother.*
Predicate Logic

• Definite descriptions (cont.)

However, there are contexts in which definite descriptions do not uniquely describe a specific given individual. A non-unique referent may be felicitously referred to by the definite NP with *the*.

Example: [Hotel concierge to a guest, in a lobby with four elevators]
“You’re in Room 611. Take *the elevator* to the sixth floor and turn left.”

Example taken from Birner & Ward (1994) “Uniqueness, Familiarity and the Definite Article in English”
Predicate Logic

• Predicates

Predicates are specified as one-place, two-place, etc., according to the number of terms they require to form a well-formed formula. The number of terms a predicate requires is called its ARITY (or ADICITY).

In English, and other natural languages, we typically find

- one-place predicates: sing, dog, happy, ...
- two-place predicates: like, know, sister (of), proud (of), ...
- three-place predicates: give, put, bring, ...

Four-place predicates are less common.
It is difficult to find good examples of five-place predicates.
Nothing prevents us from assuming such five-place predicates and higher arity predicates in artificial languages.
Predicate Logic

**Predicates**

The notion of an \( n \)-place predicate is often generalized to **zero-place predicates**. These are predicates that do not need any overt term to form a sentence. The most often cited examples are weather verbs in pro-drop languages, like

- **Latin:** \( pluit \) [rains.3Person.Singular]
- **Italian:** \( piove \)

Such Latin and Italian examples correspond to *it rains* or *it is raining* in English. However, the existence of such zero-place predicates is not uncontroversial. Some linguists (Bolinger, for example) believe that even in such cases there is an implicit term that corresponds to something like “the ambient *it*” that is absorbed by the weather predicate.
Predicate Logic

- **Predicates**
  - may appear in different forms in English:
  - The non-logical predicate constant $D$ is used to translate the (grammatical) predicate *is a dog* in the sentence *Pluto is a dog*, that is, we abstract away from copula verbs and articles.
    
    \[
    Pluto\ is\ a\ dog\ \ \ \ \ \ \ D(p)
    \]
  - In predicate logic, complex expressions like *buttered a slice of bread in the bathroom at 11 pm on Saturday* can be regarded as single one-place predicates in a logical formula:
    
    \[
    buttered\_a\_slice\_of\_bread\_in\_the\_bathroom\_at\_11pm\_on\_Saturday(j)
    \]
  - This also means that a sentence like *John broke the glass* does not have to be analyzed as a formula with a two-place predicate $B(j,g)$, but it could be also analyzed as a formula with a one-place predicate $B(j)$.
    
    \[
    John\ broke\ the\ glass\ \ \ \ \ \ \ B(j,\ g)\ \ \ \ \ B = broke,\ g = the\ glass
    \]
    
    \[
    B(j)\ \ \ \ \ B = broke\ the\ glass
    \]
Predicate Logic

*John is nicer than Peter.*

**Key**

- j: John
- p: Peter
- N(x,y): x nicer-than y (two-place predicate)

**Translation**

N(j,p)
Predicate Logic

*Peter went with Charles on Marion’s new bicycle to Amsterdam.*

Key

- p: Peter
- c: Charles
- b: Marion’s new bicycle
- a: Amsterdam

G(x,y,z,w): x go with y on z to w (four-place predicate)

Translation

G(p,c,b,a)
Predicate Logic

*Charles and Elsa are brother and sister or nephew and niece.*

**Key**

- c: Charles
- e: Elsa
- B(x,y): x and y are brother and sister
- N(x,y): x and y are nephew and niece

**Translation**

\[ B(c,e) \lor N(c,e) \]
Predicate Logic

*Marion is a happy woman.*

Key

- m: Marion
- H(x): x is happy
- W(x): x is a woman

Translation

\[ H(m) \land W(m) \]
Predicate Logic

*Charles is boring or irritating*

**Key**

- c: Charles
- B(x): x is boring
- I(x): x is irritating

**Translation**

B(c) \lor I(c)
Predicate Logic

- There is one significant difference between the way we treat predicates within the LINGUISTIC THEORY OF SYNTAX (as in Toy English) and how we deal with sentences in the SYNTAX OF PREDICATE LOGIC.
- Let us compare the sentence Alan knows Betty with the transitive verb knows in Toy English and a proposition with a two-place predicate knows in predicate logic:
**Predicate Logic**

In *Toy English*, we have a category for the result of combining a transitive verb with the direct object *NP*, namely, the *VP*, which does not exist in standard predicate logic. In this respect, the syntax of predicate logic is simpler.

In predicate logic, the combination of predicates with their arguments is referred to as a **predicate-argument structure**: $K(a,b)$.

The left–to–right order of the arguments in the predicate-argument structure corresponds to the order of subject, direct object, indirect object, and so on, in the natural language sentence:

```
John gave a rabbit to Mary.
```

<table>
<thead>
<tr>
<th>Subject</th>
<th>DO</th>
<th>IO</th>
<th>syntactic functions</th>
</tr>
</thead>
<tbody>
<tr>
<td>G (j, r, m)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The left–to–right order of individual constant terms $j$, $r$, $m$ provides the link between lexical semantics (the $n$-place nature of the predicate) and syntactic structure (the syntactic function of the terms in the sentence).
Predicate Logic

- Predicate-argument structures of FOL are generated by the first syntactic rule of predicate logic.

1. If $P$ is an $n$-place (i.e., $n$-ary) predicate constant and $t_1, ..., t_n$ are individual terms (constants or variables), then $P(t_1, ..., t_n)$ is a well-formed formula $\phi$.

- This rule also excludes combinations of a predicate with the wrong number of individual terms as not well-formed:

  If $G$ stands for the three-place predicate *give*, as in *John gave Mary a yacht*, then
  
  *$G(j,m)$* (wrong)
  
  $G(j,m,y)$ (correct)

  In English, *John gave Mary* is not well-formed either.

  *John gave Mary a yacht* is well-formed.
Predicate Logic

- **Connectives** are assigned to no category at all, since they have no meaning when standing by themselves, they only have a meaning in conjunction with other expressions.

- Connectives are called SYNCATEGOREMATIC (*syn*- ‘with’, Latin from Greek) symbols, because they are introduced into complex expressions by the formation rules *along with* the regular, or categorematic, symbols.
Predicate Logic

- **Propositional function** (term is due to Bertrand Russell, 1905)
  
  Example: *Dog(x)* ‘x is a dog’

  Propositional function is a formula with at least one free variable, here *x*. Propositional function is also called an
  - open formula,
  - open sentence,
  - open statement.

- *x* is an **individual variable**
- *x* serves as an **argument** of the propositional function *Dog(x)*

- **In general, an individual variable like *x* does not stand for a particular individual, but for any individual, for an arbitrary individual.**
- **An individual variable stands for a term whose reference is indeterminate.**
Predicate Logic

• We have seen that
  the reference of a declarative sentence is its truth value: 1, 0.

• In propositional logic, simple declarative sentences correspond to atomic propositions, and every atomic proposition can take one of two truth values: 1 or 0.

• Propositions in predicate logic are also true or false.

  Is the propositional function \( \text{Dog}(x) \) ‘x is a dog’ true or false?
Predicate Logic

• The truth value for propositional functions like $\text{Dog}(x)$ ‘x is a dog’ is indeterminate, because we do not know what their variables actually stand for.

Therefore,

• propositional functions in predicate logic cannot be said to be true or false.
Predicate Logic

- Variables in predicate logic are similar to variables in arithmetics. For example, in

\[(2 + x) = y\]

the numbers for which \(x\) and \(y\) stand are indeterminate. They stand in a certain relation to each other. If \(x\) stands for 3, \(y\) must stand for 5:

\[(2 + 3) = 5\]
Predicate Logic

When the dog named Pluto,

represented by the individual constant \( p \), is **ASSIGNED to** \( x \) (as the value of \( x \)) in the **propositional function** \( Dog(x) \), we get the proposition \( Dog(p) \).
Predicate Logic

Notice that a **propositional function is satisfied by individuals**, and not by names of individuals.

![Image of Pluto]

the individual Pluto is assigned to $x$ in

propositional function $\text{Dog}(x)$ ‘$x$ is a dog’

proposition $\text{Dog}(p)$ ‘Pluto is a dog’
Predicate Logic

In sum, we analyze the proposition that Pluto is a dog into two components:

- the individual (or argument) Pluto
- the propositional function ‘x is a dog’

\[ p + Dog(x) \]

\[ \text{proposition } \textit{Dog}(p) \quad \text{‘Pluto is a dog’} \]
Predicate Logic

• A propositional function is a function that takes different individuals as arguments and gives different propositions as values.
Predicate Logic

For different values for $x$ in $D(x)$, we may get different truth values.

- If the individual named *Pluto* is assigned to $x$, we can say that the individual Pluto satisfies the propositional function $Dog(x)$ ‘$x$ is a dog’, iff Pluto is a dog.
- This amounts to saying that $Dog(p)$ is a true proposition, whereby $p$ (the individual constant) refers to the unique individual Pluto.
- The individual Alan does not satisfy the propositional function $Dog(x)$ ‘$x$ is a dog’, because Alan is not a dog.
Predicate Logic

- **QUANTIFIERS.** A formula that is a propositional function can be made into a proposition by applying one of the following two operations to each of its variables:
  1. assign a value to the variable
  2. quantify the variable using a quantifier, i.e., prefix a formula with a quantifier that binds the variable in that formula.

- The **universal quantifier** is represented by $\forall$ and is the correspondent of English expressions such as *all, each, every.*

  \[ \forall x \ [S(x)] \]  
  ‘For every individual $x$ in the domain, it holds that $x$ has the property $S.$’

- The **existential quantifier**, represented by $\exists,$ corresponds to *some* in the sense of *at least one, possibly more.*

  \[ \exists x \ [S(x)] \]  
  ‘There is at least one individual $x$ in the domain such that $x$ has the property $S.$’
Predicate Logic

• Quantificational expression: all, some, many, two, most, every …

Introduce
• the power to express generalizations into language, i.e.,
• the power to move beyond talk about properties of particular individuals to saying what QUANTITY of the individuals in a given domain have a given property.

(Chierchia & McConnell-Ginet 1990/2000, 114ff.)
Predicate Logic

• In the interpretation of a sentence like

   \[ \text{Pluto barked} \quad B(p) \]

   the proper name \textit{Pluto} corresponds a unique individual in our domain of discourse.

• In the interpretation of quantified sentences like

   \[ \text{Every dog barked.} \]
   \[ \text{Some dog barked.} \]

   quantificational expressions like \textit{every dog} and \textit{some dog} CANNOT be represented by a unique individual in our domain of discourse, because they do not refer to a single unique individual.

   \textbf{How do we interpret such quantified sentences?}
Predicate Logic

• The interpretation of sentences with quantificational expressions is systematically related to the interpretation of sentences without quantificational expressions.
  = The basic insight of the theory of quantification.

• There is a systematic connection between the truth conditions of sentences with quantificational expressions as subjects and the truth conditions of sentences with ordinary referring expressions like proper names as subjects.

\[
\begin{array}{|c|c|}
\hline
\text{Pluto} & \text{barked.} \\
\hline
p & x \text{ barked} = B(p) \\
\hline
\text{Every dog} & \text{barked.} \\
\text{quantificational expression} & x \text{ barked} \\
\hline
\end{array}
\]

• It is this connection that the theory of quantification exploits.
• The truth conditions for quantified sentences are defined with respect to two components:
  1. quantificational expression like \textit{every dog} and
  2. propositional function (open sentence) like \( x \text{ barked} \)

(Chierchia & McConnell-Ginet 1990/2000, 114ff.)
Every dog
quantificational expression +

QUANTITY instruction
how many dogs in the model
should have the barking property

barked.
x barked

SINGULAR subject-predicate open sentence
contains an unquantified variable for
individuals and attributes to its potential
referents the property of barking

In the process of evaluating quantified sentences, we
may think of the variable x as a placeholder,
something like a SINGULAR pronoun it, he, she

(Chierchia & McConnell-Ginet 1990/2000, 114ff.)
Every dog barked.
quantified expression + x (= it) barked

• The truth-conditional import of a sentence like Every dog barked can be illustrated by uttering it barked again and again, each time accompanied by a pointing at a different dog in our model (domain of discourse) until each dog has been pointed at. Relative to each pointing, it barked will be assigned the truth value T or F: Pluto barked - true, Dean barked - true, ...

• Every dog barked is true relative to every possible pointing, i.e., it is true iff each and every dog in our model M barked.

the denotation of dog in the model M (i.e., all the dogs that there are in the world that we talk about)

it barked

(Chierchia & McConnell-Ginet 1990/2000, 114ff.)
Predicate Logic

This idea can be applied to other quantified sentences:

• Someone smiled.
• No one is bald.
• Three cats sat on a mat.

(Chierchia & McConnell-Ginet 1990/2000, 114ff.)
Predicate Logic

This idea can be applied to other quantified sentences:

Someone smiled
- can be roughly decomposed into *(Someone) (it/he/she smiled)*
- is true relative to some pointing or other
- is true just in case at least one individual in the model \( M \) smiled

it/he/she smiled

Mona Lisa smiled.

*(Chierchia & McConnell-Ginet 1990/2000, 114ff.)*
Predicate Logic

No one is bald
• can be roughly decomposed into (No one) (it/he/she is bald)
• is true relative to NO pointing
• is false just in case at least one individual
  in the model $M$ is bald

(Chierchia & McConnell-Ginet 1990/2000, 114ff.)
Predicate Logic

*Three cats sat on a mat.*
- can be roughly decomposed into *(Three cats) (it sat on a mat)*
- is true relative to (at least) 3 different pointings to cats, just in case there are at least 3 different cats to point to of whom *sat on a mat* is true

(Chierchia & McConnell-Ginet 1990/2000, 114ff.)
Predicate Logic

Summary so far: in order to understand the truth-conditional import of quantified sentences, we decompose them into two components:

Every dog barked (every dog) (it barked)  
Someone smiled. (someone) (it/he/she smiled)  
No one is bald. (no one) (it/he/she is bald)  
Three cats sat on a mat. (three cats) (it sat on a mat)

1: a SINGULAR subject-predicate sentence with a placeholder, here represented by a pronoun, as its subject, which can be evaluated as true or false for the pronoun -- demonstrated by pointing, for example -- and which by itself expresses no generalization

2: a QUANTIFICATIONAL expression, which is the generalizing or quantity component, it tells us something about how many different values of the placeholder (pronoun) we have to consider

(Chierchia & McConnell-Ginet 1990/2000, 114ff.)
Predicate Logic

Every dog barked  (every dog)  (it barked)
Someone smiled.  (someone)  (it/he/she smiled)
No one is bald.  (no one)  (it/he/she is bald)
Three cats sat on a mat.  (three cats)  (it sat on a mat)

TWO-STAGE approach to the truth-conditional analysis of quantified sentences:

1 Truth conditions are defined for the singular sentence relative to some value for the placeholder(s), here represented by a pronoun, and then
2 Truth conditions are defined in terms of generalizations (encoded by quantificational expressions) about values assigned to the singular sentence

It was Gottlob Frege (1848-1925) who proposed the two-stage approach to the analysis of quantified sentences

(Chierchia & McConnell-Ginet 1990/2000, 114ff.)
Predicate Logic

Every dog barked.
\[ \forall x[\text{Dog}(x) \rightarrow \text{Bark}(x)] \]
‘For all x, if x is a dog, then x must bark.’

This formula is true even if there are no dogs in the universe of discourse. Why?
Predicate Logic

Every dog barked.
\( \forall x [\text{Dog}(x) \rightarrow \text{Bark}(x)] \)
‘For all x, if x is a dog, then that x must bark.’

This formula is true even if there are no dogs in the universe of discourse.

• This is due to the truth-functional properties of the logical conditional connective ‘\( \rightarrow \)’ (material implication).
• If \( \text{Dog}(x) \) is false for all values of x, then \( [\text{D}(x) \rightarrow \text{B}(x)] \) will be true. Compare the truth table for the logical connective ‘\( \rightarrow \)’.

<table>
<thead>
<tr>
<th>p</th>
<th>q</th>
<th>p ( \rightarrow ) q</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>
Predicate Logic

*Some dog barked.*

\[ \exists x [\text{Dog}(x) \land \text{Bark}(x)] \]

‘There is at least one individual in the model (universe of discourse) who has both the dog property and the barking property.’

! The absence of dogs from the universe of discourse makes the proposition false. Why?
Predicate Logic

*Some dog barked.*
\[ \exists x [\text{Dog}(x) \land \text{Bark}(x)] \]
‘There is at least one individual in the model (universe of discourse) who has both the dog property and the barking property.’

- The absence of dogs from the universe of discourse makes the proposition false (presupposition failure).
- This follows from the truth-functional properties of the logical conjunction ‘\&’
  If \( \text{Dog}(x) \) is false for all values of \( x \), then \([\text{D}(x) \land \text{B}(x)]\) will be false.
- Compare the truth table for the logical connective ‘\&’:

<table>
<thead>
<tr>
<th>p</th>
<th>q</th>
<th>p &amp; q</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
Predicate Logic

• We have

Every dog barked.

\[\forall x[\text{Dog}(x) \rightarrow \text{Bark}(x)]\]

• Why is \[\forall x[\text{Dog}(x) \land \text{Bark}(x)]\] wrong?

The formula requires that every individual in the universe of discourse is both a dog and barks.

It excludes the possibility of there being other individuals with other properties in the domain of discourse.
Predicate Logic

• We have \( \exists x [\text{Dog}(x) \land \text{Bark}(x)] \)

• Why is \( \exists x [\text{Dog}(x) \rightarrow \text{Bark}(x)] \) wrong?
  
  – The formula says that there is at least one \( x \) in the universe of discourse such that if that \( x \) is a dog, then that \( x \) barks.
  
  – The formula is true even if there are no dogs in the universe of discourse, or there is at least one individual (any individual, not necessarily a dog) that barks.
  
  – Some dog barked would then get the same translation as \emph{Either there are no dogs or there is at least one barking individual}. It follows from

  \textbf{Conditional Law:} \( p \rightarrow q \iff \neg p \lor q \)

  but it is obviously not correct.

  – The formula also says that the sentence should come out true if there is ANY and at least one individual that is not a dog, regardless whether there is any individual barking (again, due to the Conditional Law.)
Predicate Logic

• **Existential Quantifier and Connective OR**

• If all the elements in the universe of discourse can be listed, then the existential quantification \( \exists x P(x) \) is equivalent to the disjunction:

\[
\exists x P(x) \iff P(x_1) \lor P(x_2) \lor P(x_3) \lor \ldots \lor P(x_n).
\]

Recall: logical equivalence amounts to truth in the same models

disjunction is true just in case at least one of its disjuncts is true

• Example: \( \exists x [\text{Brown-Dog}(x)] \)

If we knew that there were only 5 dogs in our universe of discourse, say

Pluto, Dean, Rex, Fluff, Toto

then we could also write the statement

\( \exists x [\text{Brown-Dog}(x)] \) as

Brown-Dog(Pluto) \lor Brown-Dog(Dean) \lor Brown-Dog(Rex) \lor Brown-Dog(Fluff) \lor Brown-Dog(Toto)
Predicate Logic

• **Universal Quantifier and Connective AND**

• If all the elements in the universe of discourse can be listed, then the universal quantification $\forall x P(x)$ is equivalent to the conjunction:

$$\forall x P(x) \iff P(x_1) \land P(x_2) \land P(x_3) \land \ldots \land P(x_n).$$

Recall: logical equivalence amounts to truth in the same models
conjunction is true just in case all of its conjuncts are true

• Example: $\forall x [\text{Brown-Dog}(x)]$
If we knew that there were only 5 dogs in our universe of discourse, say Pluto, Dean, Rex, Fluff, Toto, then we could also write the statement
$\forall x [\text{Brown-Dog}(x)]$ as

$\text{Brown-Dog}(\text{Pluto}) \land \text{Brown-Dog}(\text{Dean}) \land \text{Brown-Dog}(\text{Rex}) \land \text{Brown-Dog}(\text{Fluff}) \land \text{Brown-Dog}(\text{Toto})$
Predicate Logic

- *No student complained.*
  i. \[ \neg \exists x [ \text{Student}(x) \land \text{Complain}(x)] \]
  ii. \[ \forall x [ \text{Student}(x) \rightarrow \neg \text{Complain}(x)] \]

- *Not every student was happy.*
  i. \[ \exists x [ \text{Student}(x) \land \neg \text{Happy}(x)] \]
  ii. \[ \neg \forall x [ \text{Student}(x) \rightarrow \text{Happy}(x)] \]

i. and ii. are logically equivalent due to
- Conditional Law
- De Morgan Law
- Quantifier Negation Law (next slide)

- It does not matter which particular formula we choose for the translation of any natural language sentence, as long as they specify the same truth conditions for the sentence.
Predicate Logic

• **Quantifier Negation**

The quantificational system of predicate logic can be defined in terms of negation plus one quantifier.

\[
\neg \forall x [\phi(x)] \iff \exists x [\neg \phi(x)] \\
\neg \exists x [\phi(x)] \iff \forall x [\neg \phi(x)] \\
\forall x [\phi(x)] \iff \neg \exists x [\neg \phi(x)] \\
\exists x [\phi(x)] \iff \neg \forall x [\neg \phi(x)]
\]
Predicate Logic

• Quantifier Negation

\[ \neg \forall x [T(x) \rightarrow F(x)] \iff \exists x [T(x) \land \neg F(x)] \]

It is not the case for all those who are teachers that they are friendly.
(I.e., it is not the case that if somebody is a teacher, then that person is friendly.)

\[ \neg \exists x [T(x) \land F(x)] \iff \forall x [T(x) \rightarrow \neg F(x)] \]

It is not the case that some teachers are friendly.
(I.e., it is not the case that there is at least 1 teacher who is friendly.)

\[ \forall x [T(x) \rightarrow F(x)] \iff \neg \exists x [T(x) \land \neg F(x)] \]

\[ \exists x [T(x) \land F(x)] \iff \neg \forall x [T(x) \rightarrow \neg F(x)] \]

Some teachers are friendly.
Predicate Logic

Everybody loves Marion.
Predicate Logic

*Everybody loves Marion.*

Key

- m: Marion
- L(x,y): x loves y
Predicate Logic

*Everybody loves Marion.*

**Key**

- **m**: Marion
- **L(x, y)**: x loves y

L(x, m)
Predicate Logic

*Everybody loves Marion.*

Key

- \( m: \) Marion
- \( L(x,y): \) \( x \) loves \( y \)

Translation

\[ \forall x [L(x, m)] \]
Predicate Logic

Some politicians are honest.

Key

\[ P(x): \text{x is a politician} \]
\[ H(x): \text{x is honest} \]
Predicate Logic

Some politicians are honest.

Key

\[ P(x) \land H(x) \]

\[ P(x) : \ x \text{ is a politician} \]

\[ H(x) : \ x \text{ is honest} \]
Predicate Logic

*Some politicians are honest.*

**Key**

- P(x): x is a politician
- H(x): x is honest

**Translation**

\[ \exists x [P(x) \land H(x)] \]
Predicate Logic

Nobody is a politician and not ambitious.

Key

\[ P(x): \text{x is a politician} \]

\[ A(x): \text{x is ambitious} \]
Predicate Logic

Nobody is a politician and not ambitious.

Key

\[ P(x) \land \neg A(x) \]
Nobody is a politician and not ambitious.

Key

- P(x): x is a politician
- A(x): x is ambitious

Translation

\[ \neg \exists x [P(x) \land \neg A(x)] \]
Predicate Logic

*It is not the case that all ambitious people are not honest.*

Key

\[ A(x): \]  \text{x is ambitious} \\
\[ H(x): \]  \text{x is honest} \\

\[ A(x) \]
\[ \neg H(x) \]
Predicate Logic

*It is not the case that all ambitious people are not honest.*

Key

- \( A(x) \): x is ambitious
- \( H(x) \): x is honest

Translation

\[ \neg \forall x [A(x) \rightarrow \neg H(x)] \]
Predicate Logic

All blond authors are clever.

Key

\begin{align*}
\text{B}(x): & \quad \text{x is blond} \\
\text{A}(x): & \quad \text{x is author} \\
\text{C}(x): & \quad \text{x is clever}
\end{align*}
Predicate Logic

All blond authors are clever.

Key

B(x): x is blond
A(x): x is author
C(x): x is clever

blond authors: B(x) ∧ A(x)
Predicate Logic

*All blond authors are clever.*

Key

- $B(x)$: $x$ is blond
- $A(x)$: $x$ is author
- $C(x)$: $x$ is clever

Translation

$$\forall x[(B(x) \land A(x)) \rightarrow C(x)]$$
Predicate Logic

Some best-selling authors are blind.

Key

A(x): x is a best-selling author
B(x): x is blind

Translation

∃x[A(x) ∧ B(x)]
Predicate Logic

Peter is an author who has written some best-selling books.

Key

- p: Peter
- A(x): x is an author
- B(x): x is a book
- S(x): x is best-selling
- W(x,y): x has written y

Translation

\[ A(p) \land \exists x [B(x) \land W(p,x) \land S(x)] \]
Predicate Logic

John loves Mary, but Mary loves someone else.

Key

j: John
m: Mary
L(x,y): x loves y
Predicate Logic

John loves Mary, but Mary loves someone else.

Key

\[
\begin{align*}
  j &: \text{John} \\
  m &: \text{Mary} \\
  L(x,y) &: \text{x loves y}
\end{align*}
\]

\[ L(j,m) \land \]
Predicate Logic

John loves Mary, but Mary loves someone else.

Key

j: John
m: Mary
L(x,y): x loves y

Translation

L(j,m) \land \exists x (L(m,x) \land x \neq j)
Predicate Logic

*If all logicians are smart, then Alfred is smart.*
Predicate Logic

*If all logicians are smart, then Alfred is smart.*

\[ \forall x [L(x) \rightarrow S(x)] \]
Predicate Logic

_If all logicians are smart, then Alfred is smart._

\( \forall x [L(x) \rightarrow S(x)] \rightarrow \)
Predicate Logic

If all logicians are smart, then Alfred is smart.

$$\forall x [L(x) \rightarrow S(x)] \rightarrow S(A)$$
Predicate Logic

• Variable binding

Someone smiled.
\[ \exists x \ [S(x)] \]
In words: ‘There is at least one x such that it/he/she smiled.’

The \( x \) written together with one of the quantifiers indicates that the quantification is with respect to that variable in the expression which follows. This labeling is necessary since an expression may contain more than one quantifier and more than one variable. For example in

\[ \exists x \forall y [\text{Love}(x,y)] \]
‘There is at least one individual such that that individual loves everyone.’

the first position in \( \text{Love}(x,y) \) is existentially quantified and the second universally.
Predicate Logic

• **Variable binding**

A proposition of the first order predicate logic is defined as a formula that does not contain any free individual variables.

• An occurrence of a variable in a wff is **BOUND** if either a specific value is assigned to it or it is quantified.
• If an occurrence of a variable is not bound, it is called **FREE**.

\[
\text{Love}(x,y)
\]
propositional function, two **free** variables \(x, y\)

\[
\exists x \forall y [\text{Love}(x,y)]
\]
proposition:
‘There is at least one individual such that that individual loves everyone.’

\[
\exists x [\text{Love}(x, \text{Pluto})]
\]
proposition:
‘There is at least one individual such that that individual loves Pluto.’

• The set of propositions is also called the set of **CLOSED FORMULAS** of the predicate logic.
• **Both open and closed formulas (or sentences) are well-formed formulas.**
Predicate Logic

- **Vacuous quantification**
- Any quantifier plus variable may be prefixed to a formula, even when that variable does not occur within the formula:
  
  \[ \forall x[P(y)] \]  
  ‘For every \( x \), it holds that \( y \) has the property \( P \).’

- Such a formula is well-formed, what we have here is an instance of vacuous quantification: Quantification over a formula with respect to a variable that does not occur in that formula.

- The syntax allows for vacuous quantification.
- To exclude vacuous quantification, we would have to introduce another rule; as a result the set of generative rules would be more complex. Keeping vacuous quantification simplifies syntactic rules. The semantic rules will treat such vacuously quantified formulas as if the vacuous quantifier simply were not there.

  \[ \forall xK(j,m) \] will be interpreted exactly like \( K(j,m) \)  
  \[ \exists yB(x) \] will be interpreted exactly like \( B(x) \)
Predicate Logic

- **Scope of a quantifier**

  - The extent of the application of a quantifier is called the scope of the quantifier.
  - It is indicated by square brackets $[]$ immediately after the quantifier: $\forall x[P(x)]$.
  - If there are no square brackets, then the scope is understood to be the smallest wff following the quantifier:
    In $\exists xP(x)$, the variable $x$ is bound by the quantifier $\exists x$.
    In $\exists xP(x)$, $P(x)$ is the scope of the quantifier $\exists x$. 
Predicate Logic

• **Scope of a quantifier**
  Examples: the scope of each quantifier is underlined.

\[ \forall x [S(x) \land P(x)] \]
  The quantifier \( \forall x \) binds the x’s in \( S(x) \) and \( P(x) \)

\[ \forall x [S(x)] \land P(x) \]
  The quantifier \( \forall x \) binds the variable x in \( S(x) \)
  The value of x in \( P(x) \) is independent of the value of the x that is in the scope of the quantifier (\( \forall x \)).
Predicate Logic

• **Scope of a quantifier**

\[ \forall x [S(x) \rightarrow \exists y[K(x, y)]] \]

- **Scope of \( \forall x \):** \([S(x) \rightarrow \exists y[K(x, y)]]\)
- **Scope of \( \exists y \):** \([K(x, y)]\)

\[ \forall x \exists y [S(x) \rightarrow K(x, y)] \]

- **Scope of \( \forall x \):** \(\exists y[S(x) \rightarrow K(x, y)]\)
- **Scope of \( \exists y \):** \([S(x) \rightarrow K(x, y)]\)
Predicate Logic

• **Scope of a quantifier**

Examples:

- $S(x)$ occurrence of $x$ is **free**
- $\forall x S(y)$ occurrence of $y$ is **free** (vacuous quantification)
- $\forall x L(x,y)$ occurrence of $y$ is **free**
- $\exists x \forall y [L(x,y) \wedge S(x)]$ the second occurrence of $x$ is **free**
Predicate Logic

• **Scope of a quantifier**

A variable that is free in a subformula may become bound in a larger formula:

$$\exists x[Q(x) \land \forall y[P(y) \rightarrow \exists z[S(x,y,z)]]]$$

scope of $$\exists z$$: $$[S(x,y,z)]$$

x and y occur free in $$\exists z[S(x,y,z)]$$, but they are bound in a larger formula

scope of $$\exists x$$: $$[Q(x) \land \forall y[P(y) \rightarrow \exists z[S(x,y,z)]]]$$

scope of $$\forall y$$: $$[P(y) \rightarrow \exists z[S(x,y,z)]]$$
Predicate Logic

• Any occurrence of a variable in a formula is either bound or free.
• A variable may only be bound once.

In the following formula

∀x[P(x) → ∃x[M(x)]]

– the first occurrence of x in P(x) is bound by (∀x) and the second occurrence of x in M(x) is bound by (∃x) only.
– Although the intervening quantifiers assure that the first and second occurrence of x are kept distinct, it is best to avoid using the same variable letter for distinct variables. Instead of the above formula, we prefer to write

∀x[P(x) → ∃y[M(y)]]
Predicate Logic

• Within the scope of a quantifier, one and the same variable always stands for one and the same individual.

• Different occurrences of the same variable should stand for the same individual when they are bound by the same quantifier occurrence.
Predicate Logic

• Scope of a quantifier

Generally, if

x is any variable and

φ is a formula to which a quantifier is attached to produce

(∃x)φ or (∀x)φ, then we say that

φ is the scope of the attached quantifier and that

φ or any part of φ lies in the scope of that quantifier.

We also refer to the formula φ as the matrix of the formula (∃x)φ or (∀x)φ.
Predicate Logic

• Scope of a quantifier

Put in more technical terms:
An occurrence $x'$ of a variable $x$ is bound by a quantifier occurrence $Q'$ of $(\exists y)$ or $(\forall x)$ iff
(i) $x'$ is in the scope of the occurrence of $Q'$;
(ii) there is no occurrence of $Q''$ of a quantifier $(\exists x)$ or $(\forall x)$ such that $Q''$ is in the scope of $Q'$, and $x'$ is the scope of $Q''$.
That is, $Q'$ is the “closest” quantifier occurrence that binds variables $x$, in terms of scope.
Predicate Logic

• **Scope of a quantifier defined in terms of c-command**

C-command:
A node $A$ c-commands a node $B$ iff the first branching node that dominates $A$ also dominates $B$.

![Diagram showing c-command relationships]

You can find c-commanded nodes visually by following a simple rule: Go up one then down one or more, and any node you land at will be c-commanded by the node you started at.
Predicate Logic

- Scope of a quantifier defined in terms of *c-command*

The scope of a quantifier is what it c-commands (Chierchia and McConnell-Ginet 1990/2000, p. 120).

An occurrence of $x$ is bound by a quantifier $Q$ iff $Q$ is the lowest quantifier c-commanding $x$.

$$\exists x [Q(x)] \lor P(x)$$

the first occurrence of $x$ in $Q(x)$ is bound, the second occurrence in $P(x)$ is free
Predicate Logic

- **Scope of a quantifier** defined in terms of *c-command*

\[ \forall x [\exists x [Q(x)] \lor P(x)] \]

∀x binds only the second occurrence of x, while ∃x binds the first one.
Predicate Logic

- **CONSTANTS**, e.g., $b$ and $c$ in $\text{Likes}(b,c)$ are not said to be bound or free.
- Binding only applies to variables.
Predicate Logic

• There is no written homework, for next time you should read

• Obligatory
  De Swart:
  • Review 4.1.4, 4.2.2, 4.3, 4.4
  • Read 5.1 and 5.2

• Optional (Quantifiers)
  Barwise & Etchemendy, Chapter 9 - 11
Predicate Logic

Generally, a propositional function like $F(x)$ contains

- $F$ = function
- $x$ = argument

The two are combined by means of

- **functional application:**
  - applying $F$ to $x$ yields the propositional function $F(x)$.

- Or, we say that the function $F$ operates on the individual variable $x$. 