Formal Logic

Lecture 8: The Semantics of Predicate Logic (Part I)

Dr. Ioannis Votsis

ioannis.votsis@nchlonodon.ac.uk

www.votsis.org
Some Ideas from Set Theory
Sets

• A set is simply a collection of things. We call the latter their *elements* or *members*.

• If element $\alpha$ belongs to set $A$, we can express this as: $\alpha \in A$. If element $\beta$ does not belong to $A$, then we express it as: $\beta \notin A$.

• We express the contents of each set within curly brackets.

  $A = \{\text{Dee, Joe, Alex, Kate}\}$
  $B = \{\text{Ann, David, Mary, Tim}\}$

• The order in which the elements appear doesn’t matter in normal sets. We could have written $A = \{\text{Kate, Alex, Dee, Joe}\}$. 
Ordered sets

• An ordered set, a.k.a. an ‘$n$-tuple’, is a set whose members are identified by pointy brackets $<$ $>$. 

• Each member of an $n$-tuple contains exactly $n$ objects that are ordered from left to right and separated by commas.

Examples of individual members of ordered sets:

\[
\begin{align*}
\langle a, b \rangle & \neq \langle b, a \rangle \\
\langle 1, 2 \rangle & \neq \langle 2, 1 \rangle \\
\langle a, b, c \rangle & \neq \langle b, c, a \rangle & \neq & \langle a, c, b \rangle & \neq & \ldots
\end{align*}
\]

Examples of ordered sets:

\[
\begin{align*}
S_1 & = \{\langle a, b \rangle\} \\
S_2 & = \{\langle a, b \rangle, \langle b, a \rangle\} \\
S_3 & = \{\langle 1, 2, 3 \rangle, \langle 3, 2, 1 \rangle\}
\end{align*}
\]
Ordered sets and relations

• It should be clear that $n$-tuples come in pairs ($n=2$), triples ($n=3$), quadruples ($n=4$), quintuples ($n=5$), and so on.

• An $n$-tuple can be used to capture all the things to which $n$-place relations apply.

A binary, i.e. 2-place, relation contains only pairs.  
Example: ‘$x$ is smaller than $y$’  
$S_1 = \{<0, 1>, <1, 2>, <2, 3>, \ldots \}$

A ternary, i.e. 3-place, relation contains only triples. 
Example: ‘$x$ is between $y$ and $z$’  
$S_1 = \{<1, 0, 2>, <2, 1, 3>, \ldots \}$

and so on for more complex relations.

NB: 1-place relations are captured by ‘unordered’ sets.
Ordered sets: Example 1

• Actually, to specify an ordered set, we first need to specify a set from which the objects of the ordered set are drawn.

• We call this the universe (a.k.a. domain) of discourse (UD).

  UD: \{Ann, Alex, Dee, David, Tim, Mary, Joe, Kate\}
  T: \{<Tim, Alex>, <Alex, Kate>, <Kate, Joe>, <Joe, Mary>, <Mary, Ann>, <Ann, David>, <David, Dee>\}

• T may in fact be capturing the relation ‘x is taller than y’.

  NB: T contains pairs. As specified above, it relates all the objects in UD but that’s not necessary, e.g. T’: \{<Tim, Ann>\}.
Ordered sets: Example 2

• Suppose our UD consists of some boxers and we’re interested in the relations ‘x is a heavyweight’ and ‘x beat y’.

UD: {Fury, Wilder, Joshua, Klitschko, Mayweather, Pacquiao}

H: {Fury, Wilder, Joshua, Klitschko}
B: {<Fury, Wilder>, <Joshua, Klitschko>, <Mayweather, Pacquiao>}

where:
H captures the 1-place relation ‘x is a heavyweight’.
B captures the 2-place relation ‘x beat y’.
Ordered sets: Example 3

• Suppose our UD consists of a bunch of different objects and we’re interested in the relation ‘x is identical to y’.

UD: \{Ioannis, Brian, David, Christoph, Naomi, 1, 2, 3, 4, 5\}

I: \{<Ioannis, Ioannis>, <Brian, Brian>, <David, David>, <Christoph, Christoph> <Naomi, Naomi>, <1, 1>, <2, 2>, <3, 3>, <4 , 4>, <5, 5>\}

where:
I captures the 2-place relation ‘x is identical to y’.

**NB**: One of the properties of relation I is reflexivity. That’s because all objects in UD bear that relation to themselves.
The Semantics of Predicate Logic
Interpretations

• Recall that in propositional logic, we gave ‘meaning’ to whole sentences by assigning truth-values to them.

• In predicate logic, there is a similar semantic notion at play, though it is a little more complex.

• An interpretation does not rely on natural language articulations of predicates.

• Instead, set theoretical structures, or more simply structures, are employed to give ‘meaning’.
• A structure gives ‘meaning’ by providing extensions to the predicates and constants of $L_2$.

• What is an extension? It is a set of objects which satisfies the given predicate or constant.

• Here’s the extension of the predicate S which in English is expressed by ‘... is a student of NCH Logic Class 2019-2020’: $S: \{\text{Cormac, David, Eirin, Isaac, Jeremy, Max, Mohamed, Phi}\}$

• Here’s the extension of the constant i: $\{\text{Ioannis}\}$. Constants always have one object as their extension.
• An interpretation in $L_2$ is provided by an $L_2$-structure that assigns extensions to UD, predicates and constants as follows:

(1) The universe of discourse UD is assigned at least one object, i.e. it must not be empty.

(2) Each $n$-place predicate is assigned an $n$-tuple from (i.e. which ranges over) the elements of UD.

(3) Each constant is assigned one element from UD.

**NB:** An interpretation also assigns truth-values to sentences (i.e. 0-place predicates) of $L_2$. 
Interpretations: Some qualifications

• An interpretation may assign the same member of UD to one or more constants, e.g. a: 1, c: 1.

**NB:** This does justice to the fact that in natural language we use different names to denote the same individual.

• When the UD is restricted to a certain class, e.g. natural numbers, we can read the quantifiers accordingly.

**UD:** The set of natural numbers

**P:** The set of prime numbers, **S:** \{<x, y>: x is smaller than y\}

\((\forall x) (Px \rightarrow (\exists y) Syx)\)

Every natural number x is such that if x is a prime then there is at least one natural number y smaller than x.
Fixing truth-values: Example 1

UD = \{1, 2, 3, 4\}
E: \{2, 4\}
G: \{<2, 1>, <3, 1>, <4, 1>, <3, 2>, <4, 2>, <4, 3>, <1, 1>, <2, 2>, <3, 3>, <4, 4>\}
P: \{<2, 3, 4>, <1, 2, 4>\}
a: 1, b: 2, c: 3, d: 4

**True Sentences**

- Eb
- \(\neg \text{Ea}\)
- Gdc
- Pbcd

**Because**

- \(2 \in E\)
- It is false that \(1 \in E\)
- \(<4, 3> \in G\)
- \(<2, 3, 4> \in P\)

**NB:** As it so happens, G stands for ‘x is greater than or equal to y’ and P stands for ‘the sum of x, y, and z is odd’.
Fixing truth-values: Example 2

UD = \{1, 2, 3, 4\}
E: \{2, 4\}
G: \{<2, 1>, <3, 1>, <4, 1>, <3, 2>, <4, 2>, <4, 3>, <1, 1>, <2, 2>, 
    <3, 3>, <4, 4>\}
P: \{<2, 3, 4>, <1, 2, 4>\}
a: 1, b: 2, c: 3, d: 4

**False Sentences**

- Ea 1 \notin E
- \neg Ed 4 \in E
- Gac <1, 3> \notin G
- Pabc <1, 2, 3> \notin P

**Because**

- 1 \notin E
- It is true that 4 \in E
- <1, 3> \notin G
- <1, 2, 3> \notin P

**NB:** As it so happens, G stands for ‘x is greater than or equal to y’ and P stands for ‘the sum of x, y, and z is odd’.
Fixing truth-values: Example 3

UD: {Ann, Tim, Mary, Joe, Kate}
T: {<Tim, Kate>, <Kate, Joe>, <Joe, Mary>, <Mary, Ann>}
a: Ann, j: Joe, k: Kate, m: Mary, t: Tim

True or False?
Taj
¬Tmt
¬Tjm
Ttk
Taj & Ttk
¬Tmt V ¬Tjm

NB: As it so happens, T stands for ‘x is taller than y’.
Fixing truth-values: Example 3 (continued)

UD: \{Ann, Tim, Mary, Joe, Kate\}
T: \{<Tim, Kate>, <Kate, Joe>, <Joe, Mary>, <Mary, Ann>\}
a: Ann, j: Joe, k: Kate, m: Mary, t: Tim

<table>
<thead>
<tr>
<th>True or False?</th>
<th>Result</th>
<th>Because</th>
</tr>
</thead>
<tbody>
<tr>
<td>Taj</td>
<td>False</td>
<td>&lt;Ann, Joe&gt; \notin T</td>
</tr>
<tr>
<td>¬Tmt</td>
<td>True</td>
<td>It is false that &lt;Mary, Tim&gt; \in T</td>
</tr>
<tr>
<td>¬Tjm</td>
<td>False</td>
<td>It is true that &lt;Joe, Mary&gt; \in T</td>
</tr>
<tr>
<td>Ttk</td>
<td>True</td>
<td>&lt;Tim, Kate&gt; \in T</td>
</tr>
<tr>
<td>Taj &amp; Ttk</td>
<td>False</td>
<td>&lt;Ann, Joe&gt; \notin T</td>
</tr>
<tr>
<td>¬Tmt V ¬Tjm</td>
<td>True</td>
<td>It is false that &lt;Mary, Tim&gt; \in T</td>
</tr>
</tbody>
</table>

**NB**: As it so happens, T stands for ‘x is taller than y’.
Fixing truth-values: Example 4

UD: \{Tom, Dick, Harry\}
B: \{\langle Tom, Dick\rangle, \langle Tom, Harry\rangle, \langle Tom, Tom\rangle, \langle Dick, Harry\rangle\}
R: \{Dick, Harry\}
P: \{Harry\}
d: Dick, h: Harry, t: Tom

True or False?

\((\forall x)\ R_x\)
\((\forall z)\ (P_z \rightarrow R_z)\)
\((\forall x)(\forall y)\ \neg B_{xy}\)
\((\forall x)(\forall y)\ (B_{xy} \rightarrow R_y)\) \& P_h

NB: As it so happens, B stands for ‘x is the boss of y’, R stands for ‘x gets a raise’ and P stands for ‘x parties’.
Fixing truth-values: Example 4 (continued)

UD: \{Tom, Dick, Harry\}
B: \{<Tom, Dick>, <Tom, Harry>, <Tom, Tom>, <Dick, Harry>\}
R: \{Dick, Harry\}
P: \{Harry\}
d: Dick, h: Harry, t: Tom

<table>
<thead>
<tr>
<th>True or False?</th>
<th>Result</th>
<th>Because</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\forall x) Rx</td>
<td>False</td>
<td>Tom (\not\in R)</td>
</tr>
<tr>
<td>(\forall z) (Pz (\rightarrow) Rz)</td>
<td>True</td>
<td>Harry (\in P) and Harry (\in R)</td>
</tr>
<tr>
<td>(\forall x)(\forall y) \neg Bxy</td>
<td>False</td>
<td>e.g. &lt;Tom, Dick&gt; (\in B)</td>
</tr>
<tr>
<td>(\forall x)(\forall y) (Bxy \rightarrow Ry) &amp; Ph</td>
<td>False</td>
<td>&lt;Tom, Tom&gt; (\in B) and Tom (\not\in R)</td>
</tr>
</tbody>
</table>

NB: As it so happens, B stands for ‘x is the boss of y’, R stands for ‘x gets a raise’ and P stands for ‘x parties’.
Fixing truth-values: Example 5

UD: \{Tom, Dick, Harry\}
B: \{<Tom, Dick>, <Tom, Harry>, <Tom, Tom>, <Dick, Harry>\}
R: \{Dick, Harry\}
P: \{Harry\}
d: Dick, h: Harry, t: Tom

**True or False?**

(\exists z) \ Pz \& \ Rd
(\exists y)(\exists z) \ Byz
(\neg(\exists y) (Ry \& Py))
(\exists x)(\forall y) \ Bxy \lor Rt

**NB:** As it so happens, B stands for ‘x is the boss of y’, R stands for ‘x gets a raise’ and P stands for ‘x parties’.
Fixing truth-values: Example 5 (continued)

UD: \{Tom, Dick, Harry\}
B: \{<Tom, Dick>, <Tom, Harry>, <Tom, Tom>, <Dick, Harry>\}
R: \{Dick, Harry\}
P: \{Harry\}
d: Dick, h: Harry, t: Tom

<table>
<thead>
<tr>
<th>True or False?</th>
<th>Result</th>
<th>Because</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\exists z) Pz &amp; Rd</td>
<td>True</td>
<td>Harry (\in) P and Dick (\in) R</td>
</tr>
<tr>
<td>(\exists y)(\exists z) Byz</td>
<td>True</td>
<td>e.g. &lt;Tom, Dick&gt; (\in) B</td>
</tr>
<tr>
<td>(\neg(\exists y) (Ry &amp; Py))</td>
<td>False</td>
<td>It is true: Harry (\in) R and Harry (\in) P</td>
</tr>
<tr>
<td>(\exists x)(\forall y) Bxy \lor Rt</td>
<td>True</td>
<td>&lt;Tom, Dick&gt; (\in) B, &lt;Tom, Harry&gt; (\in) B and &lt;Tom, Tom&gt; (\in) B</td>
</tr>
</tbody>
</table>
The End