Overview

- Exam 1: Sentential Logic Translations (+)
- Exam 2: Sentential Logic Derivations
- Exam 3: Predicate Logic Translations
- Exam 4: Predicate Logic Derivations
- Exam 5: (finals) very similar to Exam 3
- Exam 6: (finals) very similar to Exam 4

When computing your final grade, I count your four highest scores. (A missed exam counts as a zero.)

Existential Quantifier

<table>
<thead>
<tr>
<th>original sentence</th>
<th>someone</th>
<th>is happy</th>
</tr>
</thead>
<tbody>
<tr>
<td>paraphrase</td>
<td>there is someone who is happy</td>
<td></td>
</tr>
<tr>
<td>pronoun = variable</td>
<td>there is some ( x ) (s.t.) ( x ) is happy</td>
<td></td>
</tr>
<tr>
<td>formula</td>
<td>( \exists x ) Hx</td>
<td></td>
</tr>
</tbody>
</table>
Universal Quantifier

<table>
<thead>
<tr>
<th></th>
<th>everyone</th>
<th>is happy</th>
</tr>
</thead>
<tbody>
<tr>
<td>original sentence</td>
<td>everyone is happy</td>
<td></td>
</tr>
<tr>
<td>paraphrase</td>
<td>no matter who you are you are happy</td>
<td></td>
</tr>
<tr>
<td>pronoun ⇒ variable</td>
<td>no matter who x is x is happy</td>
<td></td>
</tr>
<tr>
<td>formula</td>
<td>∀x Hx</td>
<td></td>
</tr>
</tbody>
</table>

Negative-Universal Quantifier

<table>
<thead>
<tr>
<th></th>
<th>not everyone</th>
<th>is happy</th>
</tr>
</thead>
<tbody>
<tr>
<td>original sentence</td>
<td>not everyone is happy</td>
<td></td>
</tr>
<tr>
<td>paraphrase</td>
<td>not: no matter who you are you are happy</td>
<td></td>
</tr>
<tr>
<td>pronoun ⇒ variable</td>
<td>not: no matter who x is x is happy</td>
<td></td>
</tr>
<tr>
<td>formula</td>
<td>¬∀x Hx</td>
<td></td>
</tr>
</tbody>
</table>

Negative-Existential Quantifier

<table>
<thead>
<tr>
<th></th>
<th>no one</th>
<th>is happy</th>
</tr>
</thead>
<tbody>
<tr>
<td>original sentence</td>
<td>no one is happy</td>
<td></td>
</tr>
<tr>
<td>paraphrase</td>
<td>there is no one who is happy</td>
<td></td>
</tr>
<tr>
<td>pronoun ⇒ variable</td>
<td>there is no x (s.t.) x is happy</td>
<td></td>
</tr>
<tr>
<td>formula</td>
<td>¬∃x Hx</td>
<td></td>
</tr>
</tbody>
</table>

Equivalences

<table>
<thead>
<tr>
<th>~∀xHx</th>
<th>not-everyone is happy</th>
</tr>
</thead>
<tbody>
<tr>
<td>~∃xHx</td>
<td>no-one is happy</td>
</tr>
</tbody>
</table>

| ~∀xHx   | everyone is un-happy  |
| ~∃xHx   | everyone is un-happy  |

| ~∀φ     | = | ∃ν~φ |
| ~∃φ     | = | ∀ν~φ |
new material for day 2

Quantifier Specification

<table>
<thead>
<tr>
<th>Generic Quantifier</th>
<th>versus</th>
<th>Specific Quantifier</th>
</tr>
</thead>
<tbody>
<tr>
<td>every one is H</td>
<td></td>
<td>every F is H</td>
</tr>
<tr>
<td>not-every one is H</td>
<td></td>
<td>not-every F is H</td>
</tr>
<tr>
<td>every one is un-H</td>
<td></td>
<td>every F is un-H</td>
</tr>
<tr>
<td>some one is H</td>
<td></td>
<td>some F is H</td>
</tr>
<tr>
<td>no one is H</td>
<td></td>
<td>no F is H</td>
</tr>
<tr>
<td>some one is un-H</td>
<td></td>
<td>some F is un-H</td>
</tr>
</tbody>
</table>

Example 1

original sentence: some Freshman is Happy
paraphrase: there is someone who ...

there is someone who is F and who is H
there is some x (F(x) & H(x))

DON'T FORGET PARENTHESES

Example 2

original sentence: no Freshman is Happy
paraphrase: there is no one who ...

there is no one who is F and who is H
there is no x (F(x) & H(x))

DON'T FORGET PARENTHESES
Example 3

<table>
<thead>
<tr>
<th>original sentence</th>
<th>every Freshman is Happy</th>
</tr>
</thead>
<tbody>
<tr>
<td>paraphrase</td>
<td>no matter who you are ... you</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>no matter who you are</th>
<th>IF you are F THEN you are H</th>
</tr>
</thead>
<tbody>
<tr>
<td>no matter who x is</td>
<td>IF x is F THEN x is H</td>
</tr>
</tbody>
</table>

\[ \forall x \left( Fx \rightarrow Hx \right) \]

DON’T FORGET PARENTHESES

Summary of Quantifier Specification

<table>
<thead>
<tr>
<th>everyone is H</th>
<th>( \forall x Hx )</th>
<th>every F is H</th>
<th>( \forall x (Fx \rightarrow Hx) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>not-everyone is H</td>
<td>( \sim \forall x Hx )</td>
<td>not-every F is H</td>
<td>( \sim \forall x (Fx \rightarrow Hx) )</td>
</tr>
<tr>
<td>everyone is un-H</td>
<td>( \forall x \sim Hx )</td>
<td>every F is un-H</td>
<td>( \forall x (Fx \rightarrow \sim Hx) )</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>someone is H</th>
<th>( \exists x Hx )</th>
<th>some F is H</th>
<th>( \exists (Fx &amp; Hx) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>no-one is H</td>
<td>( \sim \exists x Hx )</td>
<td>no F is H</td>
<td>( \sim \exists (Fx &amp; Hx) )</td>
</tr>
<tr>
<td>someone is un-H</td>
<td>( \exists \sim Hx )</td>
<td>some F is un-H</td>
<td>( \exists (Fx &amp; \sim Hx) )</td>
</tr>
</tbody>
</table>

Arrow versus Ampersand

Rule of Thumb (not absolute)

- the connective immediately “beneath” a universal quantifier (\( \forall \)) is usually a conditional (\( \rightarrow \))
- \( \forall (\rightarrow ) \)

- the connective immediately “beneath” an existential quantifier (\( \exists \)) is usually a conjunction (\( & \))
- \( \exists ( & ) \)

Conjunctive Predicate-Combinations

\[ x \text{ is an American Biker} = [Ax \land Bx] \]
\[ x \text{ is an AB} = [Ax \land Bx] \]
\[ \text{every AMERICAN BIKER is CLEVER} \]
\[ \forall x \left( [Ax \land Bx] \rightarrow Cx \right) \]
\[ \text{every AB is C} \]
\[ \forall x \left( [Ax \land Bx] \rightarrow Cx \right) \]
\[ \text{some AMERICAN BIKER is CLEVER} \]
\[ \exists x \left( [Ax \land Bx] \land Cx \right) \]
\[ \text{some AB is C} \]
\[ \exists x \left( [Ax \land Bx] \land Cx \right) \]
\[ \text{no AMERICAN BIKER is CLEVER} \]
\[ \forall x \left( [Ax \land Bx] \rightarrow \sim Cx \right) \]
\[ \text{no AB is C} \]
\[ \sim \exists x \left( [Ax \land Bx] \land Cx \right) \]
Non-Conjunctive Predicates

- alleged criminal
- imitation leather
- expectant mother
- experienced sailor
- small whale
- large shrimp
- deer hunter
- racecar driver

- woman racecar driver
- baby whale killer
- dandruff shampoo
- productivity software

Ambiguous Examples

- Bostonian cab driver
- Bostonian attorney

A Pitfall

Compare the following:

\[
\text{every Bostonian Attorney is Clever}
\]

\[
\text{every BA is C}
\]

\[
\text{vs.}
\]

\[
\text{every Bostonian and Attorney is Clever}
\]

\[
\text{every B and A is C}
\]

Zombie Logic

\[
\forall x \left\{ \left( C_x \land D_x \right) \rightarrow P_x \right\}
\]

\[
\text{for any thing IF it is a Cat and it is a Dog THEN it is a Pet}
\]

in other words

\[
\text{every CAT-DOG is a PET}
\]
WHAT EXACTLY IS A CAT-DOG?

“Distributive” Use of ‘And’

every Cat and Dog is a Pet
every Cat and every Dog is a Pet
every Cat is a Pet, and every Dog is a Pet

∀x (Cx → Px) ∧ ∀x (Dx → Px)

“Plural” Use of ‘And’

every member of the class Cats-and-Dogs is a Pet

no matter who x is
  if x is a member of the class Cats-and-Dogs,
  then x is a Pet

IS
  to be a member of the class Cats-and-Dogs
  to be a Cat or a Dog

= x is a member of the class Cats-and-Dogs,
  x is a Cat or x is a Dog
= [Cx ∨ Dx]

∀x ( [Cx ∨ Dx] → Px )
‘Only’ as a Quantifier

only A are B

examples

only Citizens are Voters
only Men play NFL football
employees only
members only
cars only
right turn only

only Employees are Allowed
only Members are Allowed
only Cars are Allowed
only Right turns are Allowed

One Rendering of ‘only’

only A are B
only if you are A, are you B (no matter who you are)
you are B only if you are A (no matter who you are)
x is B only if x is A (no matter who x is)
x is not B if x is not A (no matter who x is)
if x is not A, then x is not B (no matter who x is)

\[ \forall x ( \sim A x \rightarrow \sim B x ) \]

Recall ‘only if’

A only IF B
not A IF not B
IF not B THEN not A

~ B \rightarrow \sim A

‘only’ is an implicit double-negative modifier

Alternative Rendering of ‘only’

only = no non

only A are B
no non-A are B
no one who is not A is B
there is no one who is not A but who is B
there is no x ( x is not A but x is B )

\[ \sim \exists x ( \sim A x \& B x ) \]

\[ = \]

\[ \forall x ( \sim A x \rightarrow \sim B x ) \]