Mathematical Logic

Part Two
Next Time

- **First-Order Translations**
  - How do we translate from English into first-order logic?

- **Quantifier Orderings**
  - How do you select the order of quantifiers in first-order logic formulas?

- **Negating Formulas**
  - How do you mechanically determine the negation of a first-order formula?

- **Expressing Uniqueness**
  - How do we say there’s just one object of a certain type?
Recap from Last Time
Recap So Far

• A *propositional variable* is a variable that is either true or false.

• The *propositional connectives* are as follows:
  • Negation: \( \neg p \)
  • Conjunction: \( p \land q \)
  • Disjunction: \( p \lor q \)
  • Implication: \( p \rightarrow q \)
  • Biconditional: \( p \leftrightarrow q \)
  • True: \( \top \)
  • False: \( \bot \)
First-Order Logic
What is First-Order Logic?

- **First-order logic** is a logical system for reasoning about properties of objects.
- Augments the logical connectives from propositional logic with
  - *predicates* that describe properties of objects,
  - *functions* that map objects to one another, and
  - *quantifiers* that allow us to reason about multiple objects.
Some Examples
\[ \text{ Likes(You, ComicBooks) } \lor \text{ Likes(You, GoodMovies) } \lor \text{ Likes(You, AwesomeWomenInTech) } \rightarrow \text{ Likes(You, BlackPanther) } \]
These blue terms are called *constant symbols*. Unlike propositional variables, they refer to *objects*, not *propositions*.

The red things that look like function calls are called *predicates*. Predicates take objects as arguments and evaluate to true or false.

\[
\text{LessThan}(3, 5) \land \text{LessThan}(5, 10) \rightarrow \text{LessThan}(3, 10)
\]

What remains are traditional propositional connectives. Because each predicate evaluates to true or false, we can connect the truth values of predicates using normal propositional connectives.
Reasoning about Objects

- To reason about objects, first-order logic uses *predicates*.
- Examples:
  
  \[ \text{Cute(Quokka)} \]
  \[ \text{Likes(DrLee, CS103)} \]
  \[ \text{Likes(DrLee, Quokka)} \]
  \[ \neg \text{Cute(Mosquito)} \]
  \[ \neg \text{Likes(DrLee, Mosquito)} \]

- Inputting arguments to a predicate produces a proposition, which is either true or false.
First-Order Sentences

- Sentences in first-order logic can be constructed from predicates applied to objects:
  
  \[ \text{Cute}(a) \rightarrow \text{Dikdik}(a) \lor \text{Kitty}(a) \lor \text{Puppy}(a) \]

  \[ \text{Succeeds}(\text{You}) \leftrightarrow \text{Practices}(\text{You}) \]

  \[ x < 8 \rightarrow x < 137 \]

  The less-than sign is just another predicate. Binary predicates are sometimes written in *infix notation* this way.

  Numbers are not “built in” to first-order logic. They’re constant symbols just like “You” and “a” above.
Equality

- First-order logic is equipped with a special predicate $=$ that says whether two objects are equal to one another.

- Equality is a part of first-order logic, just as $\rightarrow$ and $\neg$ are.

- Examples:

  \[
  \text{TomMarvoloRiddle} = \text{LordVoldemort} \\
  \text{MorningStar} = \text{EveningStar}
  \]

- Equality can only be applied to objects; to state that two propositions are equal, use $\leftrightarrow$. 
Let's see some more examples.
\[ \text{FavoriteMovieOf(You)} \neq \text{FavoriteMovieOf(Date)} \land \text{StarOf(FavoriteMovieOf(You))} = \text{StarOf(FavoriteMovieOf(Date))} \]
\[
\text{FavoriteMovieOf}(\text{You}) \neq \text{FavoriteMovieOf}(\text{Date}) \land \\
\text{StarOf}(\text{FavoriteMovieOf}(\text{You})) = \text{StarOf}(\text{FavoriteMovieOf}(\text{Date}))
\]
\[
\text{FavoriteMovieOf(\textit{You})} \neq \text{FavoriteMovieOf(\textit{Date})} \land \\
\text{StarOf(FavoriteMovieOf(\textit{You}))} = \text{StarOf(FavoriteMovieOf(\textit{Date}))}
\]
\[ \text{FavoriteMovieOf(You)} \neq \text{FavoriteMovieOf(Date)} \land \\
\text{StarOf(FavoriteMovieOf(You))} = \text{StarOf(FavoriteMovieOf(Date))} \]
\[
\text{FavoriteMovieOf(You)} \neq \text{FavoriteMovieOf(Date)} \land \\
\text{StarOf(FavoriteMovieOf(You))} = \text{StarOf(FavoriteMovieOf(Date))}
\]

These purple terms are \textit{functions}. Functions take objects as input and produce objects as output.
\[ \text{FavoriteMovieOf(You)} \neq \text{FavoriteMovieOf(Date)} \land \]
\[ \text{StarOf(FavoriteMovieOf(You))} = \text{StarOf(FavoriteMovieOf(Date))} \]
\[
\text{FavoriteMovieOf}(You) \neq \text{FavoriteMovieOf}(Date) \land \text{StarOf}(\text{FavoriteMovieOf}(You)) = \text{StarOf}(\text{FavoriteMovieOf}(Date))
\]
\[
\text{FavoriteMovieOf}(\text{You}) \neq \text{FavoriteMovieOf}(\text{Date}) \land \\
\text{StarOf}(\text{FavoriteMovieOf}(\text{You})) = \text{StarOf}(\text{FavoriteMovieOf}(\text{Date}))
\]
Functions

• First-order logic allows functions that return objects associated with other objects.
• Examples:

  \[ \text{ColorOf}(\text{Sky}) \]
  \[ \text{MedianOf}(x, y, z) \]
  \[ x + y \]

• As with predicates, functions can take in any number of arguments, but always return a single value.
• Functions evaluate to objects, not propositions.
Objects and Predicates

- When working in first-order logic, be careful to keep objects (actual things) and predicates (true or false) separate.
- You cannot apply connectives to objects:
  \[
  \text{Venus} \rightarrow \text{TheSun}
  \]
- You cannot apply functions to propositions:
  \[
  \text{StarOf(} \text{IsRed(Sun)} \land \text{IsGreen(Mars))}
  \]
- Ever get confused? \textit{Just ask!}
The Type-Checking Table

<table>
<thead>
<tr>
<th></th>
<th>... operate on ...</th>
<th>... and produce</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connectives (\leftrightarrow, \land, \text{etc.}) ...</td>
<td>propositions</td>
<td>a proposition</td>
</tr>
<tr>
<td>Predicates (=, \text{etc.}) ...</td>
<td>objects</td>
<td>a proposition</td>
</tr>
<tr>
<td>Functions ...</td>
<td>objects</td>
<td>an object</td>
</tr>
</tbody>
</table>
Consider the following formula in first-order logic:

\[ R(y) \rightarrow (S(x, y) = T(x)) \]

Assuming that this formula is syntactically correct, which of \( R, S, \) and \( T \) are \textit{predicates} and which are \textit{functions}?

A. \( R \) is a \textit{predicate}, \( S \) is a \textit{predicate}, and \( T \) is a \textit{predicate}.
B. \( R \) is a \textit{predicate}, \( S \) is a \textit{predicate}, and \( T \) is a \textit{function}.
C. \( R \) is a \textit{predicate}, \( S \) is a \textit{function}, and \( T \) is a \textit{predicate}.
D. \( R \) is a \textit{predicate}, \( S \) is a \textit{function}, and \( T \) is a \textit{function}.
E. \( R \) is a \textit{function}, \( S \) is a \textit{predicate}, and \( T \) is a \textit{predicate}.
F. \( R \) is a \textit{function}, \( S \) is a \textit{predicate}, and \( T \) is a \textit{function}.
G. \( R \) is a \textit{function}, \( S \) is a \textit{function}, and \( T \) is a \textit{predicate}.
H. \( R \) is a \textit{function}, \( S \) is a \textit{function}, and \( T \) is a \textit{function}.

Answer at \textbf{PollEv.com/cs103} or text \textbf{CS103} to \textbf{22333} once to join, then A, B, C, ..., or H.
A new twist: how many?
Some muggle is intelligent.
Some muggle is intelligent.

\[ \exists m. (Muggle(m) \land \text{Intelligent}(m)) \]
Some muggle is intelligent.

∃m. (Muggle(m) ∧ Intelligent(m))

∃ is the *existential quantifier* and says “for some choice of m, the following is true.”
The Existential Quantifier

- A statement of the form

\[ \exists x. \text{some-formula} \]

is true if, for some choice of \( x \), the statement \text{some-formula} is true when that \( x \) is plugged into it.

- Examples:

\[ \exists x. (\text{Even}(x) \land \text{Prime}(x)) \]
\[ \exists x. (\text{TallerThan}(x, \text{me}) \land \text{LighterThan}(x, \text{me})) \]
\[ (\exists w. \text{Will}(w)) \rightarrow (\exists x. \text{Way}(x)) \]
The Existential Quantifier

\[ \exists x. \text{Smiling}(x) \]
The Existential Quantifier

$\exists x. \text{Smiling}(x)$
The Existential Quantifier

$$\exists x. Smiling(x)$$
The Existential Quantifier

∃x. Smiling(x)
The Existential Quantifier

$$\exists x. \text{Smiling}(x)$$
The Existential Quantifier

∃x. Smiling(x)
The Existential Quantifier

∃x. Smiling(x)
The Existential Quantifier

\[ \exists x. \text{Smiling}(x) \]

Since \( \text{Smiling}(x) \) is true for some choice of \( x \), this statement evaluates to true.

Since \( \text{Smiling}(x) \) is true for some choice of \( x \), this statement evaluates to true.
The Existential Quantifier

The symbol $\exists$ represents the existential quantifier. When we say $\exists x. \text{Smiling}(x)$, we mean that there exists at least one $x$ for which the property $\text{Smiling}(x)$ is true. Since $\text{Smiling}(x)$ is true for some choice of $x$, this statement evaluates to true.
The Existential Quantifier

∃x. Smiling(x)
The Existential Quantifier

∃x. Smiling(x)
The Existential Quantifier

$\exists x. \text{Smiling}(x)$
The Existential Quantifier

$\exists x. Smiling(x)$
The Existential Quantifier

\[ \exists x. \text{Smiling}(x) \]
The Existential Quantifier

\[ \exists x. \text{Smiling}(x) \]
The Existential Quantifier

$\exists x. Smiling(x)$
The Existential Quantifier

\[ \exists x. \text{Smiling}(x) \]

Since \( \text{Smiling}(x) \) is not true for any choice of \( x \), this statement evaluates to false.
The Existential Quantifier

\[ \exists x. \text{Smiling}(x) \]

Since \( \text{Smiling}(x) \) is not true for any choice of \( x \), this statement evaluates to false.

Since \( \text{Smiling}(x) \) is not true for any choice of \( x \), this statement evaluates to false.
The Existential Quantifier

In this world, this first-order logic statement is...

A. ... true.
B. ... false.
C. ... neither true nor false.

(∃x. Smiling(x)) → (∃y. WearingHat(y))
The Existential Quantifier

$(\exists x. \text{Smiling}(x)) \rightarrow (\exists y. \text{WearingHat}(y))$
The Existential Quantifier

Is this part of the statement true or false?

$$(\exists x. \text{Smiling}(x)) \rightarrow (\exists y. \text{WearingHat}(y))$$
The Existential Quantifier

Is this part of the statement true or false?

\((\exists x. \text{Smiling}(x)) \rightarrow (\exists y. \text{WearingHat}(y))\)
The Existential Quantifier

\[(\exists x. \text{Smiling}(x)) \rightarrow (\exists y. \text{WearingHat}(y))\]

Is this part of the statement true or false?
The Existential Quantifier

(∃x. Smiling(x)) → (∃y. WearingHat(y))

Is this part of the statement true or false?
The Existential Quantifier

\[ (\exists x. \text{Smiling}(x)) \rightarrow (\exists y. \text{WearingHat}(y)) \]

Is this overall statement true or false?
The Existential Quantifier

\(\exists x. \text{Smiling}(x) \rightarrow \exists y. \text{WearingHat}(y)\)

Is this overall statement true or false?
Fun with Edge Cases

∃x. Smiling(x)
Fun with Edge Cases

Existentially-quantified statements are false in an empty world, since it’s not possible to choose an object!

\[ \exists x. \text{Smiling}(x) \]
Some Technical Details
Variables and Quantifiers

• Each quantifier has two parts:
  • the variable that is introduced, and
  • the statement that's being quantified.
• The variable introduced is scoped just to the statement being quantified.

\[(\exists x. \text{Loves}(\text{You, } x)) \land (\exists y. \text{Loves}(y, \text{You}))\]
Variables and Quantifiers

• Each quantifier has two parts:
  • the variable that is introduced, and
  • the statement that's being quantified.

• The variable introduced is scoped just to the statement being quantified.

(∃x. Loves(You, x)) ∧ (∃y. Loves(y, You))

The variable \(x\) just lives here.
The variable \(y\) just lives here.
Variables and Quantifiers

• Each quantifier has two parts:
  • the variable that is introduced, and
  • the statement that's being quantified.
• The variable introduced is scoped just to the statement being quantified.

(∃x. Loves(You, x)) ∧ (∃y. Loves(y, You))
Variables and Quantifiers

Each quantifier has two parts:

- the variable that is introduced, and
- the statement that's being quantified.

The variable introduced is scoped just to the statement being quantified.

\[(\exists x. Loves(You, x)) \land (\exists x. Loves(x, You))\]
Variables and Quantifiers

- Each quantifier has two parts:
  - the variable that is introduced, and
  - the statement that's being quantified.
- The variable introduced is scoped just to the statement being quantified.

\[(\exists x. \text{Loves}(\text{You}, x)) \land (\exists x. \text{Loves}(x, \text{You}))\]
Operator Precedence (Again)

• When writing out a formula in first-order logic, quantifiers have precedence just below \( \neg \).

• The statement

\[
\exists x. P(x) \land R(x) \land Q(x)
\]

is parsed like this:

\[
(\exists x. P(x)) \land (R(x) \land Q(x))
\]

• This is syntactically invalid because the variable \( x \) is out of scope in the back half of the formula.

• To ensure that \( x \) is properly quantified, explicitly put parentheses around the region you want to quantify:

\[
\exists x. (P(x) \land R(x) \land Q(x))
\]
“For any natural number $n$, 
$n$ is even iff $n^2$ is even”
“For any natural number \( n \),
\( n \) is even iff \( n^2 \) is even”

\( \forall n. (n \in \mathbb{N} \rightarrow (Even(n) \leftrightarrow Even(n^2))) \)
“For any natural number $n$, $n$ is even iff $n^2$ is even”

\( \forall n. \ (n \in \mathbb{N} \rightarrow (\text{Even}(n) \leftrightarrow \text{Even}(n^2))) \)

\( \forall \) is the **universal quantifier** and says “for any choice of $n$, the following is true.”
The Universal Quantifier

- A statement of the form
  \[ \forall x. \text{some-formula} \]
  is true if, for every choice of \(x\), the statement \text{some-formula} is true when \(x\) is plugged into it.

- Examples:
  \[ \forall p. (\text{Puppy}(p) \rightarrow \text{Cute}(p)) \]
  \[ \text{Tallest} (\text{SultanKösen}) \rightarrow \forall x. (\text{SultanKösen} \neq x \rightarrow \text{ShorterThan}(x, \text{SultanKösen})) \]
The Universal Quantifier

∀x. Smiling(x)
The Universal Quantifier

∀x. Smiling(x)
The Universal Quantifier

∀x. Smiling(x)
The Universal Quantifier

∀x. Smiling(x)
The Universal Quantifier

∀x. Smiling(x)
The Universal Quantifier

∀x. Smiling(x)
The Universal Quantifier

$\forall x. \text{Smiling}(x)$
The Universal Quantifier

∀x. Smiling(x)

Since Smiling(x) is true for every choice of x, this statement evaluates to true.
The Universal Quantifier

\[ \forall x. \text{Smiling}(x) \]

Since \text{Smiling}(x) is true for every choice of \( x \), this statement evaluates to true.
The Universal Quantifier

∀x. Smiling(x)
The Universal Quantifier

∀x. Smiling(x)
The Universal Quantifier

∀x. Smiling(x)
The Universal Quantifier

∀x. Smiling(x)
The Universal Quantifier

\( \forall x. \text{Smiling}(x) \)

Since \( \text{Smiling}(x) \) is false for this choice \( x \), this statement evaluates to false.
The Universal Quantifier

\[ \forall x. \text{Smiling}(x) \]

Since \( \text{Smiling}(x) \) is false for this choice of \( x \), this statement evaluates to false.
The Universal Quantifier

\[(\forall x. \text{Smiling}(x)) \rightarrow (\forall y. \text{WearingHat}(y))\]
The Universal Quantifier

\[(\forall x. \text{Smiling}(x)) \rightarrow (\forall y. \text{WearingHat}(y))\]
The Universal Quantifier

\((\forall x. \text{Smiling}(x)) \rightarrow (\forall y. \text{WearingHat}(y))\)

Is this part of the statement true or false?
The Universal Quantifier

(∀x. Smiling(x)) → (∀y. WearingHat(y))

Is this part of the statement true or false?
The Universal Quantifier

\[(\forall x. \text{Smiling}(x)) \rightarrow (\forall y. \text{WearingHat}(y))\]

Is this part of the statement true or false?
The Universal Quantifier

Is this part of the statement true or false?

\((\forall x. \text{Smiling}(x)) \rightarrow (\forall y. \text{WearingHat}(y))\)
The Universal Quantifier

\((\forall x. \text{Smiling}(x)) \rightarrow (\forall y. \text{WearingHat}(y))\)

Is this overall statement true or false in this scenario?
The Universal Quantifier

(∀x. Smiling(x)) → (∀y. WearingHat(y))

Is this overall statement true or false in this scenario?
Fun with Edge Cases

∀x. Smiling(x)
Universally-quantified statements are \textit{vacuously true} in empty worlds.

$\forall x. \text{Smiling}(x)$
Time-Out for Announcements!
Translating into First-Order Logic
Translating Into Logic

- First-order logic is an excellent tool for manipulating definitions and theorems to learn more about them.
- Need to take a negation? Translate your statement into FOL, negate it, then translate it back.
- Want to prove something by contrapositive? Translate your implication into FOL, take the contrapositive, then translate it back.
Using the predicates

- *Puppy*(p), which states that *p* is a puppy, and
- *Cute*(x), which states that *x* is cute,

write a sentence in first-order logic that means “all puppies are cute.”

Which of these first-order logic statements is a proper translation?

A. \( \exists p. (\text{Puppy}(p) \land \text{Cute}(p)) \)
B. \( \exists p. (\text{Puppy}(p) \rightarrow \text{Cute}(p)) \)
C. \( \forall p. (\text{Puppy}(p) \land \text{Cute}(p)) \)
D. \( \forall p. (\text{Puppy}(p) \rightarrow \text{Cute}(p)) \)
E. More than one of these.
F. None of these.

Answer at PollEv.com/cs103 or text CS103 to 22333 once to join, then A, B, C, D, E, or F.
An Incorrect Translation

All puppies are cute!

\[ \forall x. (\text{Puppy}(x) \land \text{Cute}(x)) \]
An Incorrect Translation

All puppies are cute!

\( \forall x. (\text{Puppy}(x) \land \text{Cute}(x)) \)

This should work for any choice of \( x \), including things that aren't puppies.
An Incorrect Translation

All puppies are cute!

∀x. (Puppy(x) ∧ Cute(x))

This should work for any choice of x, including things that aren't puppies.
An Incorrect Translation

All puppies are cute!

∀x. (Puppy(x) ∧ Cute(x))

This should work for any choice of x, including things that aren't puppies.
An Incorrect Translation

All puppies are cute!

∀x. (Puppy(x) ∧ Cute(x))

This should work for any choice of x, including things that aren't puppies.
An Incorrect Translation

All puppies are cute!

∀x. (Puppy(x) ∧ Cute(x))

This should work for any choice of x, including things that aren't puppies.
An Incorrect Translation

All puppies are cute!

\[ \forall x. (\text{Puppy}(x) \land \text{Cute}(x)) \]

This should work for any choice of x, including things that aren't puppies.
An Incorrect Translation

All puppies are cute!

∀x. \((\text{Puppy}(x) \land \text{Cute}(x))\)

A statement of the form

∀x. something

is true only when something is true for every choice of x.
An Incorrect Translation

All puppies are cute!

\[ \forall x. \ (Puppy(x) \land Cute(x)) \]

A statement of the form \( \forall x. \text{something} \) is true only when \text{something} is true for every choice of \( x \).
An Incorrect Translation

All puppies are cute!

\[ \forall x. \ (\text{Puppy}(x) \land \text{Cute}(x)) \]
An Incorrect Translation

All puppies are cute!

\( \forall x. (\text{Puppy}(x) \land \text{Cute}(x)) \)

This first-order statement is false even though the English statement is true. Therefore, it can't be a correct translation.
An Incorrect Translation

All puppies are cute!

$\forall x. \ (Puppy(x) \land Cute(x))$

The issue here is that this statement asserts that everything is a puppy. That's too strong of a claim to make.
A Better Translation

All puppies are cute!

∀x. (Puppy(x) → Cute(x))
A Better Translation

All puppies are cute!

\[ \forall x. \ (Puppy(x) \rightarrow Cute(x)) \]

This should work for any choice of \( x \), including things that aren't puppies.
A Better Translation

All puppies are cute!

∀x. (Puppy(x) → Cute(x))

This should work for any choice of x, including things that aren't puppies.
A Better Translation

All puppies are cute!

∀x. (Puppy(x) → Cute(x))

This should work for any choice of x, including things that aren't puppies.
A Better Translation

All puppies are cute!

∀x. (Puppy(x) → Cute(x))

This should work for any choice of x, including things that aren't puppies.
A Better Translation

All puppies are cute!

∀x. (Puppy(x) → Cute(x))

This should work for any choice of x, including things that aren't puppies.
A Better Translation

All puppies are cute!

∀x. (Puppy(x) → Cute(x))

This should work for any choice of x, including things that aren't puppies.
A Better Translation

All puppies are cute!

∀x. (Puppy(x) → Cute(x))

This should work for any choice of x, including things that aren't puppies.
A Better Translation

All puppies are cute!

∀x. (Puppy(x) → Cute(x))

This should work for any choice of x, including things that aren't puppies.
A Better Translation

All puppies are cute!

∀x. (Puppy(x) → Cute(x))

This should work for any choice of x, including things that aren't puppies.
A Better Translation

All puppies are cute!

∀x. (Puppy(x) → Cute(x))

This should work for any choice of x, including things that aren't puppies.
A Better Translation

All puppies are cute!

∀x. (Puppy(x) → Cute(x))

This should work for any choice of x, including things that aren't puppies.
A Better Translation

All puppies are cute!

\( \forall x. (\text{Puppy}(x) \rightarrow \text{Cute}(x)) \)

This should work for any choice of \( x \), including things that aren't puppies.
A Better Translation

All puppies are cute!

\[ \forall x. (\text{Puppy}(x) \rightarrow \text{Cute}(x)) \]

This should work for any choice of \(x\), including things that aren't puppies.
A Better Translation

All puppies are cute!

\[ \forall x. (\text{Puppy}(x) \rightarrow \text{Cute}(x)) \]

This should work for any choice of \( x \), including things that aren't puppies.
A Better Translation

All puppies are cute!

∀x. (Puppy(x) → Cute(x))

This should work for any choice of x, including things that aren't puppies.
All puppies are cute!

∀x. (Puppy(x) → Cute(x))

This should work for any choice of x, including things that aren't puppies.
A Better Translation

All puppies are cute!

\( \forall x. (\text{Puppy}(x) \rightarrow \text{Cute}(x)) \)

This should work for any choice of \( x \), including things that aren't puppies.
A Better Translation

All puppies are cute!

∀x. (Puppy(x) → Cute(x))

This should work for any choice of x, including things that aren't puppies.
A Better Translation

All puppies are cute!

∀x. \( \text{Puppy}(x) \rightarrow \text{Cute}(x) \)

This should work for any choice of \( x \), including things that aren't puppies.
A Better Translation

All puppies are cute!

∀x. (Puppy(x) → Cute(x))

This should work for any choice of x, including things that aren't puppies.
A Better Translation

All puppies are cute!

∀x. (Puppy(x) → Cute(x))

This should work for any choice of x, including things that aren't puppies.
A Better Translation

All puppies are cute!

∀x. (Puppy(x) → Cute(x))

This should work for any choice of x, including things that aren't puppies.
A Better Translation

All puppies are cute!

∀x. (Puppy(x) → Cute(x))
A Better Translation

All puppies are cute!

∀x. (Puppy(x) → Cute(x))

A statement of the form

∀x. something

is true only when something is true for every choice of x.
A Better Translation

All puppies are cute!

∀x. (Puppy(x) → Cute(x))

A statement of the form

∀x. something

is true only when something is true for every choice of x.
“All $P$'s are $Q$'s”

translates as

$\forall x. (P(x) \rightarrow Q(x))$
Useful Intuition:

Universally-quantified statements are true unless there's a counterexample.

$$\forall x. \ (P(x) \rightarrow Q(x))$$

If $x$ is a counterexample, it must have property $P$ but not have property $Q.$
Using the predicates

- *Blobfish*(b), which states that *b* is a blobfish, and
- *Cute*(x), which states that *x* is cute,

write a sentence in first-order logic that means “some blobfish is cute.”
Using the predicates

- $Blobfish(b)$, which states that $b$ is a blobfish, and
- $Cute(x)$, which states that $x$ is cute,

write a sentence in first-order logic that means “some blobfish is cute.”

Which of these first-order logic statements is a proper translation?

A. $\exists b. (Blobfish(b) \land Cute(b))$
B. $\exists b. (Blobfish(b) \rightarrow Cute(b))$
C. $\forall b. (Blobfish(b) \land Cute(b))$
D. $\forall b. (Blobfish(b) \rightarrow Cute(b))$
E. More than one of these.
F. None of these.

Answer at PollEv.com/cs103 or text CS103 to 22333 once to join, then A, B, C, D, E, or F.
An Incorrect Translation

Some blobfish is cute.

$\exists x. (\text{Blobfish}(x) \rightarrow \text{Cute}(x))$
An Incorrect Translation

Some blobfish is cute.

∃x. (Blobfish(x) → Cute(x))
An Incorrect Translation

Some blobfish is cute.

∃x. (Blobfish(x) → Cute(x))
An Incorrect Translation

Some blobfish is cute.

\[ \exists x. (\text{Blobfish}(x) \rightarrow \text{Cute}(x)) \]
An Incorrect Translation

Some blobfish is cute.

$\exists x. (Blobfish(x) \rightarrow Cute(x))$
An Incorrect Translation

Some blobfish is cute.

∃x. (Blobfish(x) → Cute(x))
An Incorrect Translation

Some blobfish is cute.

\[ \exists x. (Blobfish(x) \rightarrow Cute(x)) \]

A statement of the form

\[ \exists x. \text{something} \]

is true only when \text{something} is true for \text{at least one} choice of \( x \).
An Incorrect Translation

Some blobfish is cute.

∃x. (Blobfish(x) → Cute(x))

A statement of the form

∃x. something

is true only when something is true for

at least one choice of x.
An Incorrect Translation

Some blobfish is cute.

∃x. (Blobfish(x) → Cute(x))
An Incorrect Translation

Some blobfish is cute.

∃x. (Blobfish(x) → Cute(x))

This first-order statement is true even though the English statement is false. Therefore, it can't be a correct translation.
An Incorrect Translation

Some blobfish is cute.

$$\exists x. \ (Blobfish(x) \rightarrow Cute(x))$$

The issue here is that implications are true whenever the antecedent is false. This statement “accidentally” is true because of what happens when x isn't a blobfish.
A Correct Translation

Some blobfish is cute.

\[ \exists x. (\text{Blobfish}(x) \land \text{Cute}(x)) \]
A Correct Translation

Some blobfish is cute.

∃x. (Blobfish(x) ∧ Cute(x))
A Correct Translation

Some blobfish is cute.

\[ \exists x. (\text{Blobfish}(x) \land \text{Cute}(x)) \]
A Correct Translation

Some blobfish is cute.

∃x. \( Blobfish(x) \land Cute(x) \)
A Correct Translation

Some blobfish is cute.

∃x. \( \text{Blobfish}(x) \land \text{Cute}(x) \)
A Correct Translation

Some blobfish is cute.

∃x. (Blobfish(x) ∧ Cute(x))
A Correct Translation

Some blobfish is cute.

∃x. (Blobfish(x) ∧ Cute(x))
A Correct Translation

Some blobfish is cute.

\( \exists x. (\text{Blobfish}(x) \land \text{Cute}(x)) \)
A Correct Translation

Some blobfish is cute.

∃x. (Blobfish(x) ∧ Cute(x))
A Correct Translation

Some blobfish is cute.

∃x. (Blobfish(x) ∧ Cute(x))
A Correct Translation

Some blobfish is cute.

∃x. (Blobfish(x) ∧ Cute(x))
A Correct Translation

Some blobfish is cute.

∃x. (Blobfish(x) ∧ Cute(x))
A Correct Translation

Some blobfish is cute.

∃x. (Blobfish(x) ∧ Cute(x))
A Correct Translation

Some blobfish is cute.

\[ \exists x. \ (\text{Blobfish}(x) \land \text{Cute}(x)) \]
A Correct Translation

Some blobfish is cute.

∃x. (Blobfish(x) ∧ Cute(x))
A Correct Translation

Some blobfish is cute.

∃x. (Blobfish(x) ∧ Cute(x))
A Correct Translation

Some blobfish is cute.

∃x. (Blobfish(x) ∧ Cute(x))
A Correct Translation

Some blobfish is cute.

$\exists x. (\text{Blobfish}(x) \land \text{Cute}(x))$
A Correct Translation

Some blobfish is cute.

$\exists x. (Blobfish(x) \land Cute(x))$
A Correct Translation

Some blobfish is cute.

∃x. (Blobfish(x) ∧ Cute(x))
A Correct Translation

Some blobfish is cute.

∃x. (Blobfish(x) ∧ Cute(x))
A Correct Translation

Some blobfish is cute.

∃x. (Blobfish(x) ∧ Cute(x))

A statement of the form

∃x. something

is true only when something is true for

at least one choice of x.
A Correct Translation

Some blobfish is cute.

\( \exists x. (\text{Blobfish}(x) \land \text{Cute}(x)) \)

A statement of the form \( \exists x. \text{something} \) is true only when \text{something} is true for at least one choice of \( x \).
“Some $P$ is a $Q$” translates as

$\exists x. (P(x) \land Q(x))$
Useful Intuition:

Existentially-quantified statements are false unless there's a positive example.

$$\exists x. (P(x) \land Q(x))$$

If $x$ is an example, it must have property $P$ on top of property $Q$. 
Good Pairings

• The \( \forall \) quantifier *usually* is paired with \( \rightarrow \).
  \[ \forall x. \, (P(x) \rightarrow Q(x)) \]

• The \( \exists \) quantifier *usually* is paired with \( \land \).
  \[ \exists x. \, (P(x) \land Q(x)) \]

• In the case of \( \forall \), the \( \rightarrow \) connective prevents the statement from being *false* when speaking about some object you don't care about.

• In the case of \( \exists \), the \( \land \) connective prevents the statement from being *true* when speaking about some object you don't care about.