Mathematical Logic

Part Two

A Note on the Career Fair

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: $p \wedge q$
  - Disjunction: $p \vee q$
  - Implication: $p \to q$
  - Biconditional: $p \leftrightarrow q$
  - True: $\top$
  - False: $\bot$

Logical Equivalence

- Two propositional formulas φ and ψ are called ***equivalent*** if they have the same truth tables.
- We denote this by writing $\varphi \equiv \psi$.
- Some examples:
  - $\neg(p \wedge q) \equiv \neg p \vee \neg q$
  - $\neg(p \vee q) \equiv \neg p \wedge \neg q$
  - $\neg p \vee q \equiv p \rightarrow q$
  - $p \wedge \neg q \equiv \neg(p \rightarrow q)$

Take out a sheet of paper!

What's the truth table for the $\rightarrow$ connective?

What's the negation of $p \to q$?

New Stuff!

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, and
  - ***functions*** that map objects to one another,
  - ***quantifiers*** that allow us to reason about multiple objects simultaneously.

Some Examples

$$Likes(You, Eggs) \land Likes(You, Tomato) \rightarrow Likes(You, Shakshuka)$$

$$AreNetflixingAndChilling(You, Him) \land \neg WatchesNetflix(You) \land \neg WatchesNetflix(Him)$$

$$IsCurmugeon(DrHouse) \leftrightarrow \neg IsHappy(DrHouse)$$

$$Likes(You, Eggs) \land Likes(You, Tomato) \rightarrow Likes(You, Shakshuka)$$

$$AreNetflixingAndChilling(You, Him) \land \neg WatchesNetflix(You) \land \neg WatchesNetflix(Him)$$

$$IsCurmugeon(DrHouse) \leftrightarrow \neg IsHappy(DrHouse)$$

$$Likes(You, Eggs) \land Likes(You, Tomato) \rightarrow Likes(You, Shakshuka)$$

$$AreNetflixingAndChilling(You, Him) \land \neg WatchesNetflix(You) \land \neg WatchesNetflix(Him)$$

$$IsCurmugeon(DrHouse) \leftrightarrow \neg IsHappy(DrHouse)$$

These blue terms are called *constant symbols*. Unlike propositional variables, they refer to *objects*, not propositions.

$$Likes(You, Eggs) \land Likes(You, Tomato) \rightarrow Likes(You, Shakshuka)$$

$$AreNetflixingAndChilling(You, Him) \land \neg WatchesNetflix(You) \land \neg WatchesNetflix(Him)$$

$$IsCurmugeon(DrHouse) \leftrightarrow \neg IsHappy(DrHouse)$$

$$Likes(You, Eggs) \land Likes(You, Tomato) \rightarrow Likes(You, Shakshuka)$$

$$AreNetflixingAndChilling(You, Him) \land \neg WatchesNetflix(You) \land \neg WatchesNetflix(Him)$$

$$IsCurmugeon(DrHouse) \leftrightarrow \neg IsHappy(DrHouse)$$

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

$$Likes(You, Eggs) \land Likes(You, Tomato) \rightarrow Likes(You, Shakshuka)$$

$$AreNetflixingAndChilling(You, Him) \land \neg WatchesNetflix(You) \land \neg WatchesNetflix(Him)$$

$$IsCurmugeon(DrHouse) \leftrightarrow \neg IsHappy(DrHouse)$$

$$Likes(You, Eggs) \land Likes(You, Tomato) \to Likes(You, Shakshuka)$$

$$AreNetflixingAndChilling(You, Him) \land \neg WatchesNetflix(You) \land \neg WatchesNetflix(Him)$$

$$IsCurmugeon(DrHouse) \leftrightarrow \neg IsHappy(DrHouse)$$

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:
  - *ExtremelyCute*(*Quokka*)
  - *DeadlockEachOther*(*Democrats*, *Republicans*)
- Predicates can take any number of arguments, but each predicate has a fixed number of arguments (called its ***arity***).
  - The arity and meaning of each predicate are typically specified in advance.
- Applying a predicate to arguments 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:

$$LikesToEat(V, M) \land Near(V, M) \to WillEat(V, M)$$

$$Cute(t) \to Dikdik(t) \lor Kitty(t) \lor Puppy(t)$$

$$x < 8 \to x < 137$$

The notation $\boldsymbol{x < 8}$ is just a shorthand for something like ***LessThan(x, 8)***. Binary predicates in math are often written like this, but symbols like **<** are not a part of first-order logic.

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 $\to$ and $\neg$ are.
- Examples:

$$MorningStar = EveningStar$$

$$TomMarvoloRiddle = LordVoldemort$$

- Equality can only be applied to ***objects***; to see if ***propositions*** are equal, use $\leftrightarrow$.

For notational simplicity, define $\neq$ as

$$x \neq y \quad \equiv \quad \neg(x = y)$$

Let's see some more examples.

$$FavoriteMovieOf(You) \neq FavoriteMovieOf(Her) \land$$
$$StarOf(FavoriteMovieOf(You)) = StarOf(FavoriteMovieOf(Her))$$

$$FavoriteMovieOf(You) \neq FavoriteMovieOf(Her) \land$$
$$StarOf(FavoriteMovieOf(You)) = StarOf(FavoriteMovieOf(Her))$$

$$FavoriteMovieOf(You) \neq FavoriteMovieOf(Her) \wedge$$
$$StarOf(FavoriteMovieOf(You)) = StarOf(FavoriteMovieOf(Her))$$

$$FavoriteMovieOf(You) \neq FavoriteMovieOf(Her) \wedge$$
$$StarOf(FavoriteMovieOf(You)) = StarOf(FavoriteMovieOf(Her))$$

$$\mathit{FavoriteMovieOf}(\mathit{You}) \neq \mathit{FavoriteMovieOf}(\mathit{Her}) \land$$
$$\mathit{StarOf}(\mathit{FavoriteMovieOf}(\mathit{You})) = \mathit{StarOf}(\mathit{FavoriteMovieOf}(\mathit{Her}))$$

These purple terms are *functions*. Functions take objects as input and produce objects as output.

$$FavoriteMovieOf(You) \neq FavoriteMovieOf(Her) \wedge$$
$$StarOf(FavoriteMovieOf(You)) = StarOf(FavoriteMovieOf(Her))$$

$$\mathit{FavoriteMovieOf}(\mathit{You}) \neq \mathit{FavoriteMovieOf}(\mathit{Her}) \land$$
$$\mathit{StarOf}(\mathit{FavoriteMovieOf}(\mathit{You})) = \mathit{StarOf}(\mathit{FavoriteMovieOf}(\mathit{Her}))$$

$$
\mathit{FavoriteMovieOf}(\mathit{You}) \neq \mathit{FavoriteMovieOf}(\mathit{Her}) \land
$$
$$
\mathit{StarOf}(\mathit{FavoriteMovieOf}(\mathit{You})) = \mathit{StarOf}(\mathit{FavoriteMovieOf}(\mathit{Her}))
$$

Functions

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

$$LengthOf(path)$$

$$MedianOf(x, y, z)$$

$$x + y$$

- As with predicates, functions can take in any number of arguments, but each function has a fixed arity.
  - As with predicates, the arity and interpretation of functions are specified in advance.
- Functions evaluate to ***objects***, not ***propositions***.
- There is no syntactic way to distinguish functions and predicates; you'll have to look at how they're used.

Objects and Predicates

- When working in first-order logic, be careful to keep objects (actual things) and predicates (true or false) separate.
- You can apply functions or predicates to objects, but cannot apply connectives to objects.
- You can apply connectives to predicates, but cannot apply connectives to objects.
- Therefore, the following aren't allowed:

$$Venus \to TheSun$$

$$LengthOf(IsRed(Sun) \land IsGreen(Mars))$$

- Ever get confused? *Just ask!*

One last (and major) change

$$\forall x.\ (IsPuppy(x) \rightarrow IsAdorable(x))$$

$$\exists a.\ (IsAdorable(a) \land \neg IsPuppy(a))$$

$$\forall x.\ (IsPuppy(x) \to IsAdorable(x))$$

$$\exists a.\ (IsAdorable(a) \land \neg IsPuppy(a))$$

$$\forall x.\ (IsPuppy(x) \to IsAdorable(x))$$

$$\exists a.\ (IsAdorable(a) \land \neg IsPuppy(a))$$

$$\forall x.\ (\mathit{IsPuppy}(x) \to \mathit{IsAdorable}(x))$$

$$\exists a.\ (\mathit{IsAdorable}(a) \land \neg \mathit{IsPuppy}(a))$$

$$\forall x.\ (IsPuppy(x) \to IsAdorable(x))$$

$$\exists a.\ (IsAdorable(a) \land \neg IsPuppy(a))$$

$$\forall x.\ (\mathit{IsPuppy}(x) \to \mathit{IsAdorable}(x))$$

$$\exists a.\ (\mathit{IsAdorable}(a) \land \neg \mathit{IsPuppy}(a))$$

These green terms are called *quantifiers*. The symbol ∀ is read "for all," and the symbol ∃ is read "there exists"

$$\forall x.\ (IsPuppy(x) \to IsAdorable(x))$$

$$\exists a.\ (IsAdorable(a) \land \neg IsPuppy(a))$$

The teal terms are *variables*. Each quantifier introduces a variable that it "quantifies over."

$$\forall x.\ (IsPuppy(x) \to IsAdorable(x))$$

$$\exists a.\ (IsAdorable(a) \land \neg IsPuppy(a))$$

The teal terms are *variables*. Each quantifier introduces a variable that it "quantifies over."

In the first case, we're saying that ***for any*** choice of x, if x is a puppy, then x is adorable.

$$\forall x.\ (IsPuppy(x) \to IsAdorable(x))$$

$$\exists a.\ (IsAdorable(a) \land \neg IsPuppy(a))$$

The teal terms are *variables*. Each quantifier introduces a variable that it "quantifies over."

In the first case, we're saying that ***for any*** choice of x, if x is a puppy, then x is adorable.

In the second, we're saying that ***there exists*** some choice of *a* where *a* is adorable and *a* is not a puppy.

$$\forall x.\ (IsPuppy(x) \to IsAdorable(x))$$

(*"All puppies are adorable"*)

$$\exists a.\ (IsAdorable(a) \land \neg IsPuppy(a))$$

(*"Something is adorable but not a puppy"*)

Quantifiers

- The biggest change from propositional logic to first-order logic is the use of ***quantifiers***.
- A ***quantifier*** is a statement that expresses that some property is true for some or all choices that could be made.
- Useful for statements like “for every action, there is an equal and opposite reaction.”

“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”

$$\forall n.\ (n \in \mathbb{N} \to (Even(n) \leftrightarrow Even(n^2)))$$

“For any natural number n,
n is even iff n^2 is even”

$$\forall n.\ (n \in \mathbb{N} \to (Even(n) \leftrightarrow Even(n^2)))$$

∀ is the ***universal quantifier***
and says “for any choice of n,
the following is true.”

The Universal Quantifier

- A statement of the form $\mathbf{\forall x.\ \psi}$ asserts that for ***every*** choice of x, the statement ψ is true when we plug in that choice of x.
- Examples:

$$\forall v.\ (\mathit{Puppy}(v) \to \mathit{Cute}(v))$$

$$\forall x.\ (\mathit{IsMillennial}(x) \to \mathit{IsSpecial}(x))$$

$$\mathit{Tallest}(\mathit{SK}) \to \\ \quad \forall x.\ (\mathit{SK} \neq x \to \mathit{ShorterThan}(x, \mathit{SK}))$$

Some muggles are intelligent.

Some muggles are intelligent.

$$\exists m.\ (Muggle(m) \land Intelligent(m))$$

Some muggles are intelligent.

$$\boxed{\exists m.}\ (Muggle(m) \land Intelligent(m))$$

∃ is the ***existential quantifier*** and says "for some choice of m, the following is true."

The Existential Quantifier

- A statement of the form $\exists x.\ \psi$ asserts that ***there is some*** choice of x where ψ is true when we plug in that x.
- Examples:

$$\exists x.\ (Even(x) \land Prime(x))$$

$$\exists x.\ (TallerThan(x, me) \land LighterThan(x, me))$$

$$(\exists x.\ Appreciates(x, me)) \to Happy(me)$$

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.

$$(\forall x.\ Loves(You, x)) \to (\forall y.\ Loves(y, You))$$

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.

$$(\forall x.\ Loves(You, x)) \to (\forall y.\ Loves(y, You))$$

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.

$$(\forall x.\ Loves(You, x)) \to (\forall y.\ Loves(y, You))$$

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.

$$(\forall x.\ \mathit{Loves}(\mathit{You}, x)) \to (\forall y.\ \mathit{Loves}(y, \mathit{You}))$$

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.

$$(\forall x.\ Loves(You, x)) \to (\forall y.\ Loves(y, You))$$

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.

$$(\forall x.\ Loves(You, x)) \to (\forall y.\ Loves(y, You))$$

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.

$$(\underbrace{\forall x.\ Loves(You, x)}_{\text{The variable } x \text{ just lives here.}}) \to (\underbrace{\forall y.\ Loves(y, You)}_{\text{The variable } y \text{ just lives here.}})$$

Operator Precedence (Again)

- When writing out a formula in first-order logic, the quantifiers ∀ and ∃ have precedence just below ¬.
- The statement

$$\forall x.\ P(x) \lor R(x) \to Q(x)$$

is parsed like this:

$$\big(\ (\forall x.\ P(x)\ )\ \ \lor\ \ R(x)\ \big)\ \ \to\ \ Q(x)$$

- 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:

$$\forall x.\ (P(x) \lor R(x) \to Q(x))$$

Time-Out for Announcements!

Checkpoints Graded

- The PS1 checkpoint assignment has been graded.
- Review your feedback on GradeScope. Contact the staff (via Piazza or by stopping by office hours) if you have any questions.
- Some technical notes:
  - Make sure that when you submit a problem set question, you mark all of the pages containing your answer. Otherwise, we might not see your whole submission!
  - If you're having any problems with GradeScope, please email the staff list.
- Best of luck on the rest of the problem set!

oSTEM Mixer

- Stanford's Chapter of oSTEM (Out in STEM) is hosting a fall mixer event on Wednesday, October 14 from 6PM – 8PM at the LGBT CRC.
- They also have a mentorship program; click **this link** for more details.

Your Questions

"Why did you choose to teach here at Stanford rather than go do CS in industry after your education? Or are there plans to do that down the line? ☺"

A couple of reasons:

1. Job satisfaction
2. High impact
3. This stuff is exciting!

I might go do something else down the line - I do think about this - but I'm really happy here now.

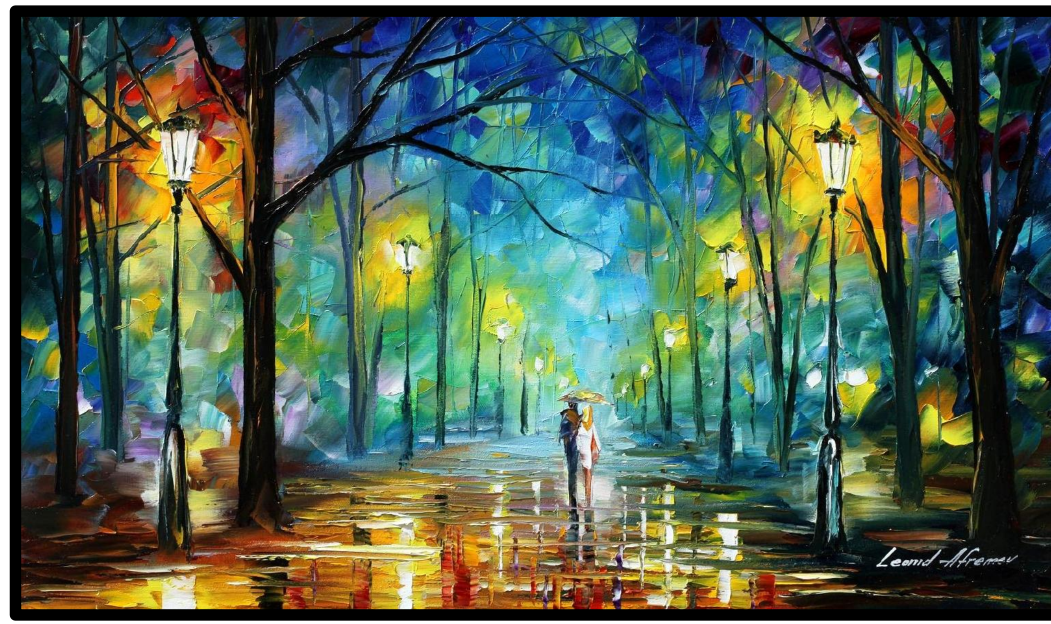
"What is your favorite area of CS and why?"

Personally, I think that theoretical CS and algorithms are really fascinating. It's amazing to try to figure out what the limits of computing are and how to reach those limits. That's just me personally, though, so please don't read anything deeper into it.

Although I don't have a huge background in it, there are a lot of cool concepts from machine learning that seem like straight-up science fiction.

Neural Artwork

Content
Result



Style

Read **[this paper](#)** for more details.

Or check out **[this github repo](#)** by a Stanford student! (Or both!)

"What CS elective at Stanford do you think is the most fun?"

That's a tough one, especially since one person's elective is another person's requirement. ☺

A sampling of elective courses I think are really fun:

CS143 CS154 CS224W CS227B CS231N CS255 CS261

Oh, and CS166. ☺

Back to CS103!

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.
- Applications:
  - Determining the negation of a complex statement.
  - Figuring out the contrapositive of a tricky implication.

Translating Into Logic

- ***Translating statements into first-order logic is a lot more difficult than it looks.***
- There are a lot of nuances that come up when translating into first-order logic.
- We'll cover examples of both good and bad translations into logic so that you can learn what to watch for.
- We'll also show lots of examples of translations so that you can see the process that goes into it.

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.”

An Incorrect Translation

All puppies are cute!

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

An Incorrect Translation

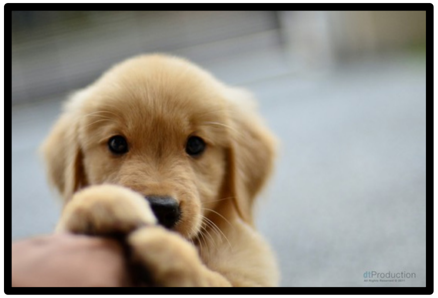
All puppies are cute!

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

This should work for <u>*any*</u> choice of x, including things that aren't puppies.

An Incorrect Translation

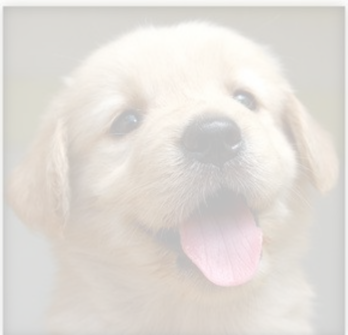
All puppies are cute!



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

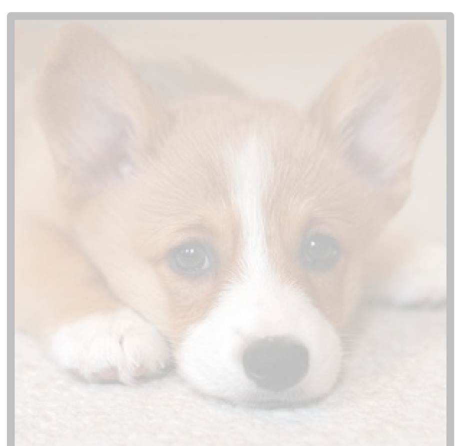
This should work for *any* choice of x, including things that aren't puppies.

An Incorrect Translation



All puppies are cute!

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

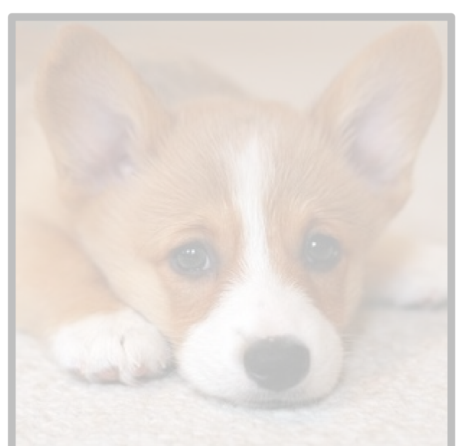


This should work for <u>any</u> choice of x, including things that aren't puppies.

An Incorrect Translation

All puppies are cute!

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



This should work for <u>*any*</u> choice of x, including things that aren't puppies.

An Incorrect Translation

All puppies are cute!

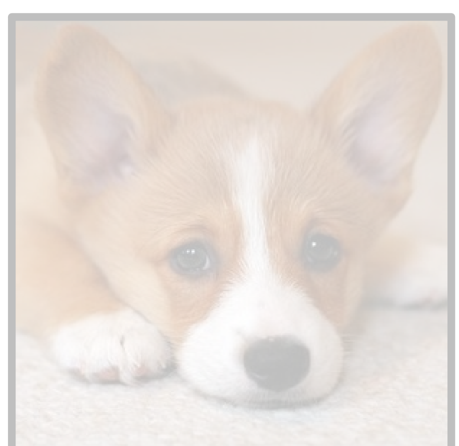
$$\forall x.\ (\cancel{\mathit{Puppy}(x)} \land \mathit{Cute}(x))$$

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

An Incorrect Translation

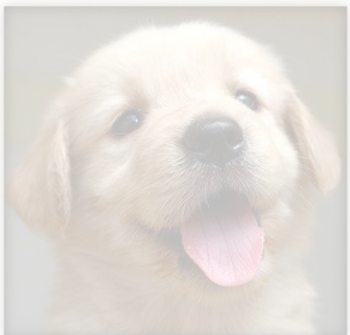
All puppies are cute!

$$\forall x.\ (\cancel{Puppy(x) \land Cute(x)})$$



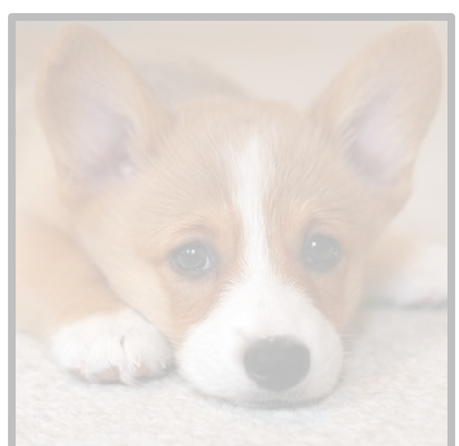
This should work for <u>*any*</u> choice of x, including things that aren't puppies.

An Incorrect Translation



All puppies are cute!

$\forall x.\ (\cancel{Puppy(x) \land Cute(x)})$



A statement of the form

$$\forall x.\ \psi$$

is true only when ψ is true for <u>every</u> choice of x.

An Incorrect Translation

All puppies are cute!

$\cancel{\forall x.\ (Puppy(x) \land Cute(x))}$

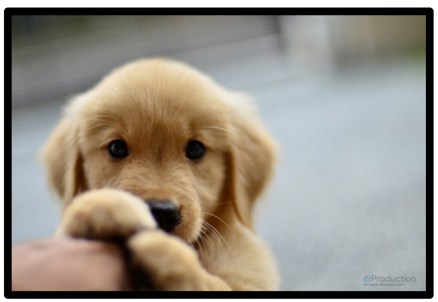
A statement of the form

$$\forall x.\ \psi$$

is true only when ψ is true for <u>every</u> choice of x.

An Incorrect Translation

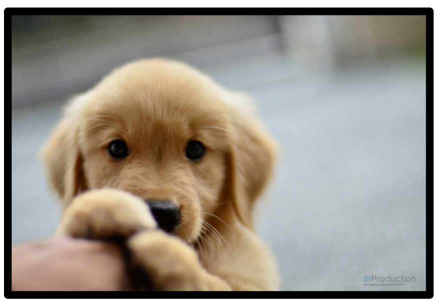
All puppies are cute!



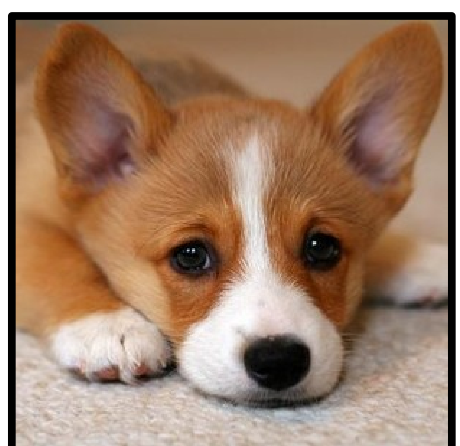
$\cancel{\forall x.\ (Puppy(x) \land Cute(x))}$

An Incorrect Translation

All puppies are cute!



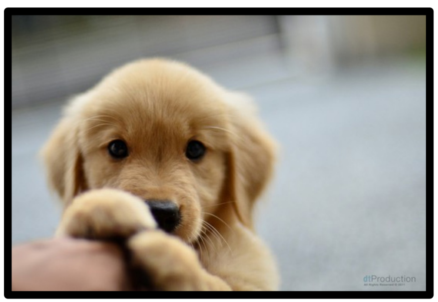
$\cancel{\forall x. (Puppy(x) \land Cute(x))}$



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!

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

A Better Translation

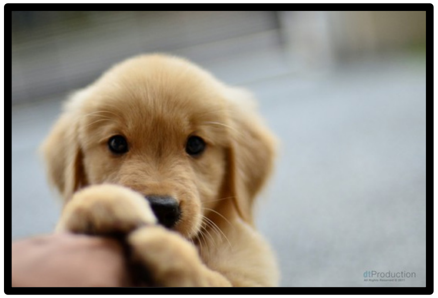
All puppies are cute!

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

This should work for <u>*any*</u> choice of x, including things that aren't puppies.

A Better Translation

All puppies are cute!



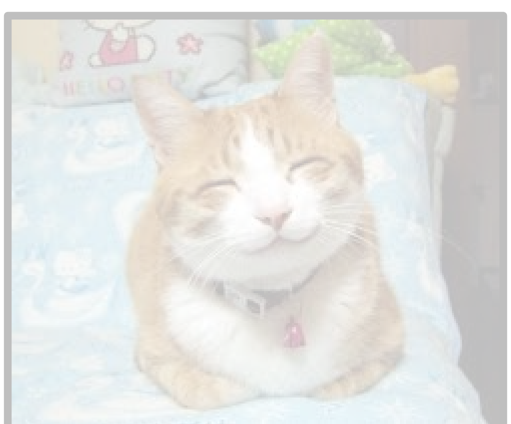
$\forall x.\ (Puppy(x) \to Cute(x))$

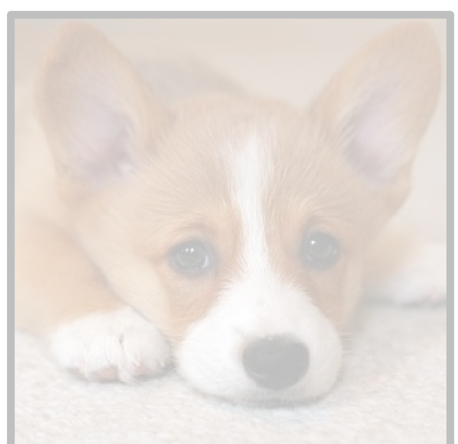
This should work for *any* choice of x, including things that aren't puppies.

A Better Translation

All puppies are cute!

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





This should work for <u>*any*</u> choice of x, including things that aren't puppies.

A Better Translation

All puppies are cute!

$\forall x. \; (Puppy(x) \to Cute(x))$

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

A Better Translation

All puppies are cute!

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

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

A Better Translation

All puppies are cute!

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

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

A Better Translation

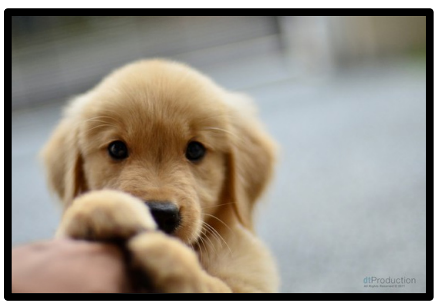
All puppies are cute!

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

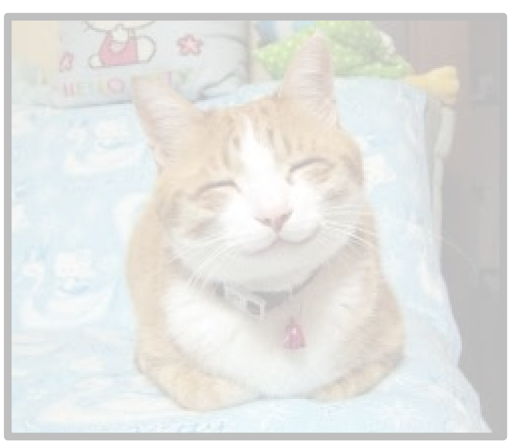
This should work for <u>*any*</u> choice of x, including things that aren't puppies.

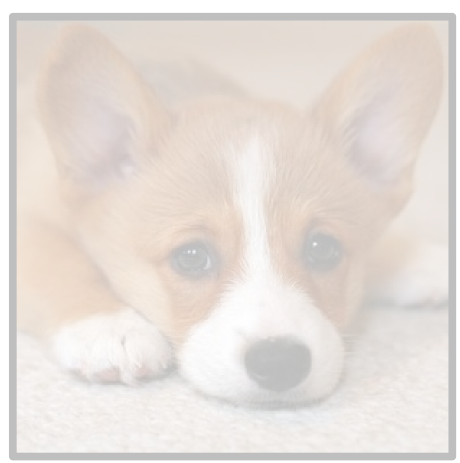
A Better Translation

All puppies are cute!



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

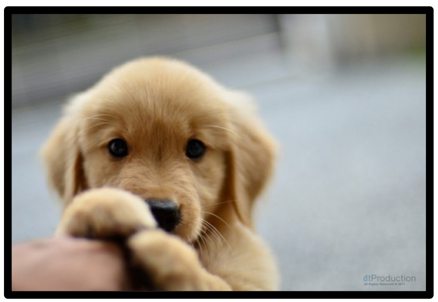




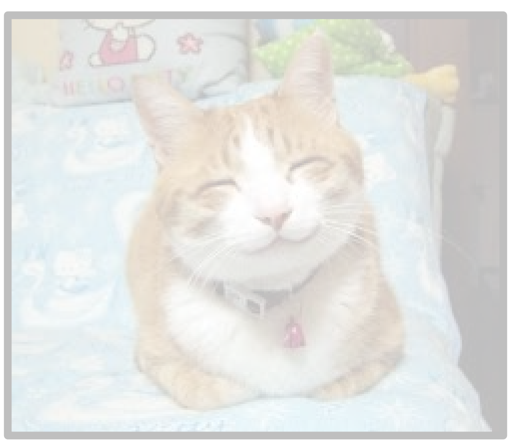
This should work for *any* choice of x, including things that aren't puppies.

A Better Translation

All puppies are cute!

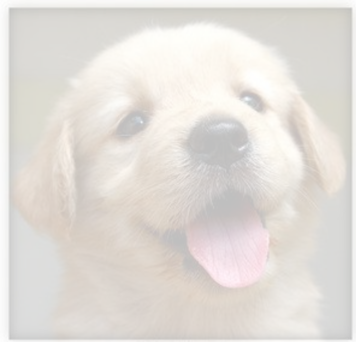


$$\forall x.\ (Puppy(x) \to Cute(x))$$

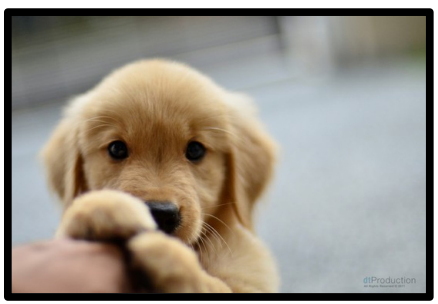


This should work for *any* choice of x, including things that aren't puppies.

A Better Translation



All puppies are cute!

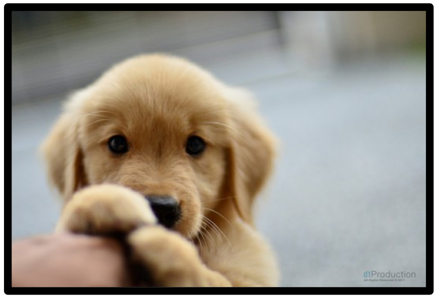


$\forall x. \, (Puppy(x) \to Cute(x))$

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

A Better Translation

All puppies are cute!

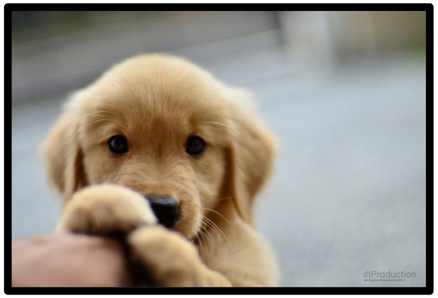


$\forall x.\ (Puppy(x) \to Cute(x))$

This should work for <u>*any*</u> choice of x, including things that aren't puppies.

A Better Translation

All puppies are cute!



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

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

A Better Translation

All puppies are cute!

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

This should work for <u>any</u> choice of x, including things that aren't puppies.

A Better Translation

All puppies are cute!

$$\forall x. (\cancel{Puppy(x)} \to Cute(x))$$

This should work for <u>*any*</u> choice of x, including things that aren't puppies.

A Better Translation

All puppies are cute!

$$\forall x.\ (\cancel{Puppy(x)} \to Cute(x))$$

This should work for <u>*any*</u> choice of x, including things that aren't puppies.

A Better Translation

All puppies are cute!

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

This should work for <u>any</u> choice of x, including things that aren't puppies.

A Better Translation

All puppies are cute!

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

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

A Better Translation

All puppies are cute!

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

This should work for <u>any</u> choice of x, including things that aren't puppies.

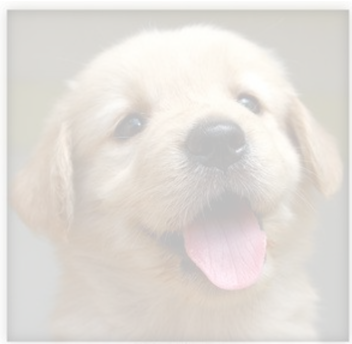
A Better Translation

All puppies are cute!

$$\forall x.\ (\mathit{Puppy}(x) \to \mathit{Cute}(x))$$

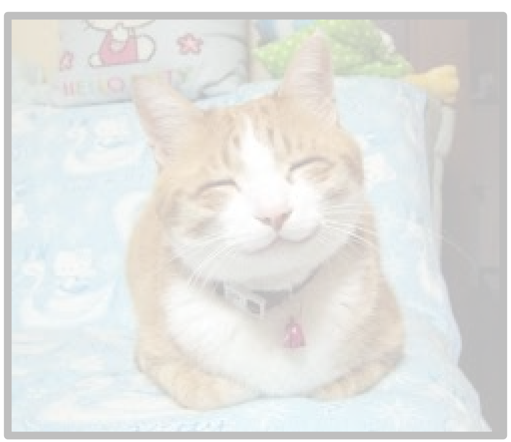
This should work for <u>any</u> choice of x, including things that aren't puppies.

A Better Translation



All puppies are cute!

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



This should work for <u>any</u> choice of x, including things that aren't puppies.

A Better Translation

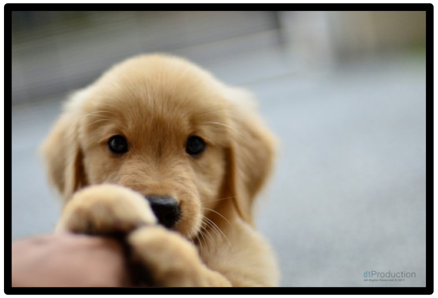
All puppies are cute!

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

This should work for <u>any</u> choice of x, including things that aren't puppies.

A Better Translation

All puppies are cute!



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

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

"All *P*'s are *Q*'s"

translates as

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

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

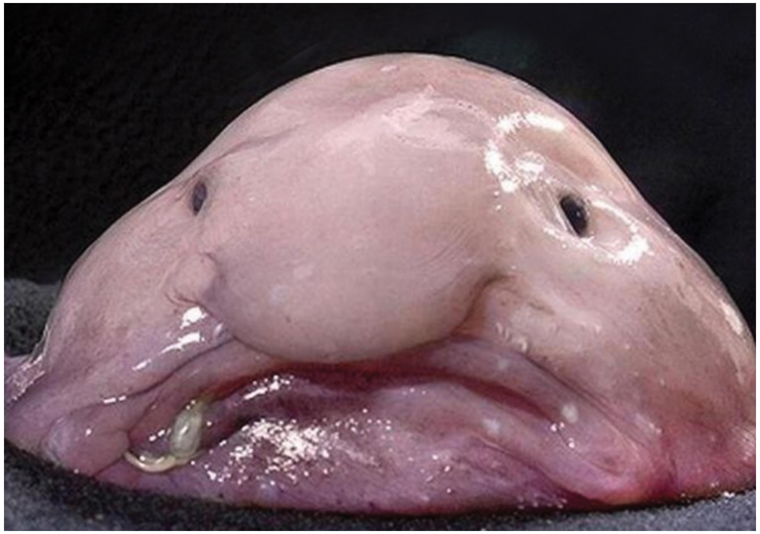
- $Blobfish(b)$, which states that b is a blobfish, and
- $Cute(x)$, which states that x is cute,

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.”



Using the predicates

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

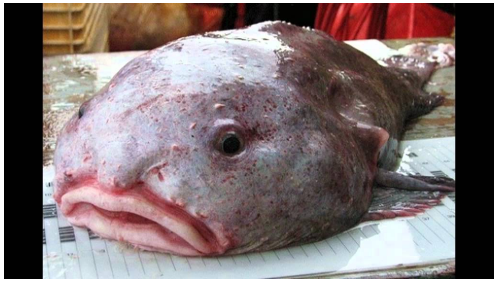
write a sentence in first-order logic that means “some blobfish is cute.”

An Incorrect Translation

Some blobfish is cute.

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

An Incorrect Translation



Some blobfish is cute.

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

An Incorrect Translation

Some blobfish is cute.

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

An Incorrect Translation

Some blobfish is cute.

$$\exists x.\ (\cancel{Blobfish(x)} \to Cute(x))$$

An Incorrect Translation

Some blobfish is cute.

$$\exists x.\ (\cancel{Blobfish(x)} \rightarrow Cute(x))$$

An Incorrect Translation

Some blobfish is cute.

$\exists x.\ (\mathit{Blobfish}(x) \to \mathit{Cute}(x))$

An Incorrect Translation

Some blobfish is cute.

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

A statement of the form

$$\exists x.\ \psi$$

is true only when ψ is true for *some* choice of x.

An Incorrect Translation

Some blobfish is cute.

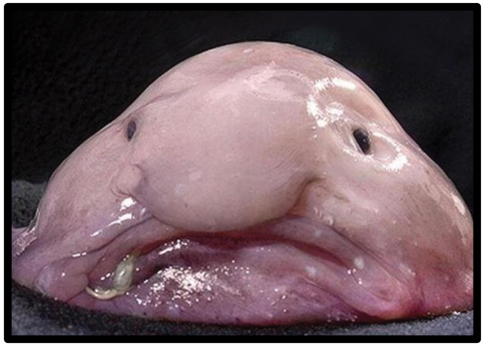
$\exists x.\ (Blobfish(x) \to Cute(x))$

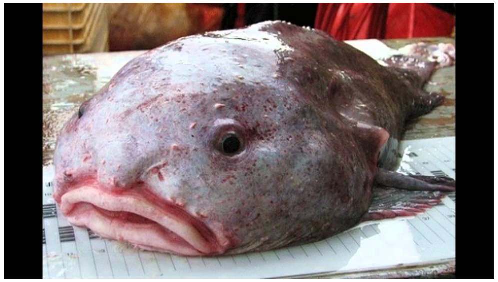
A statement of the form

$\exists x.\ \psi$

is true only when ψ is true for *some* choice of x.

An Incorrect Translation

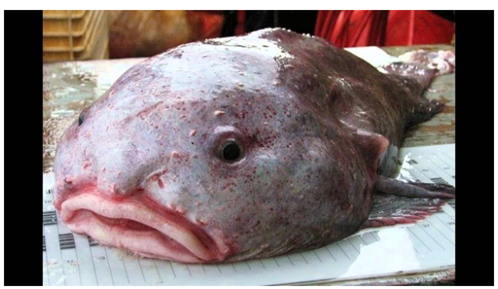




Some blobfish is cute.

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

An Incorrect Translation



Some blobfish is cute.

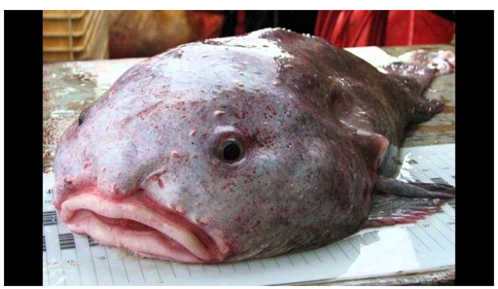
$$\exists x.\ (Blobfish(x) \rightarrow Cute(x))$$

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.

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



The issue here is that implications are true whenever the antecedent is false. Using an implication here lets this be true for the wrong reason.

A Correct Translation

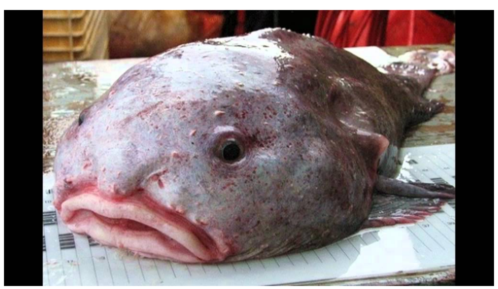
Some blobfish is cute.

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

A Correct Translation

Some blobfish is cute.

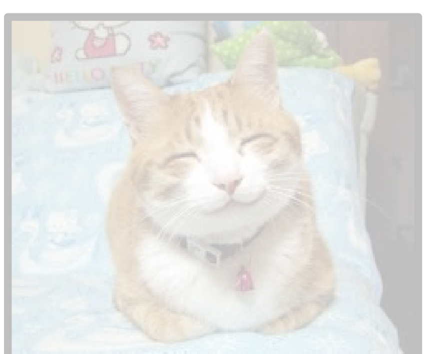
$\exists x.\ (Blobfish(x) \land Cute(x))$



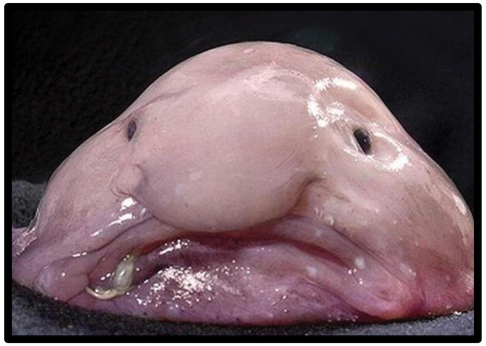
A Correct Translation

Some blobfish is cute.

$\exists x. \; (Blobfish(x) \land Cute(x))$



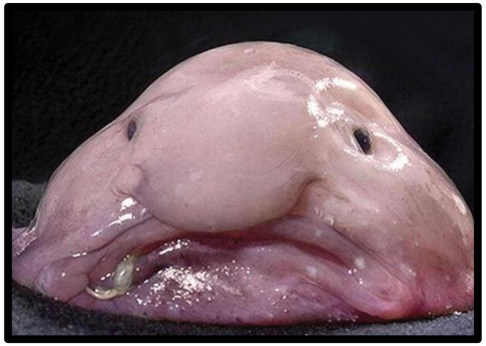
A Correct Translation



Some blobfish is cute.

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

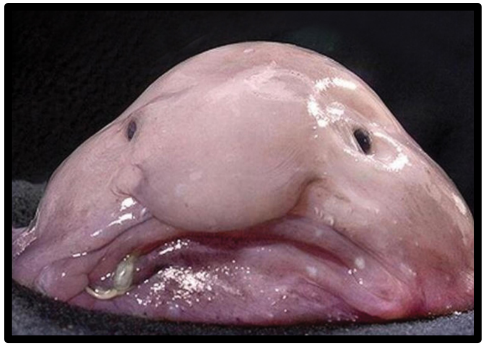
A Correct Translation



Some blobfish is cute.

$$\exists x. \, (\mathit{Blobfish}(x) \land \cancel{\mathit{Cute}(x)})$$

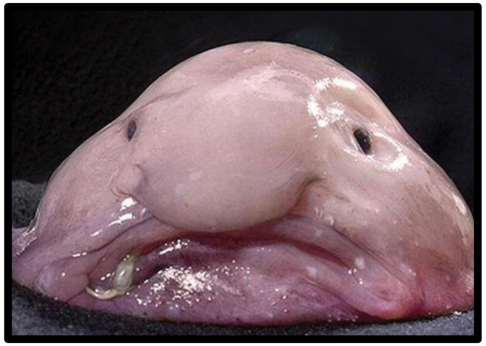
A Correct Translation



Some blobfish is cute.

$\exists x. \; (\cancel{Blobfish(x) \land Cute(x)})$

A Correct Translation



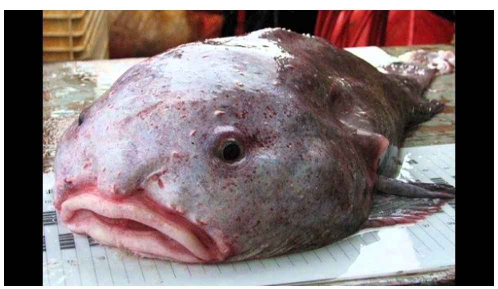
Some blobfish is cute.

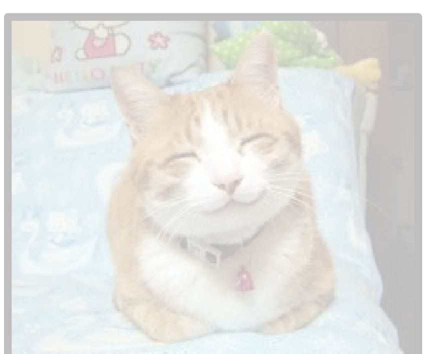
$\exists x. (Blobfish(x) \land Cute(x))$

A Correct Translation

Some blobfish is cute.

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

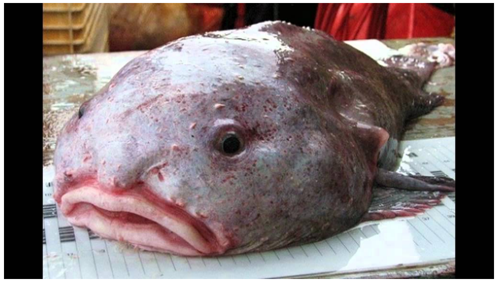




A Correct Translation

Some blobfish is cute.

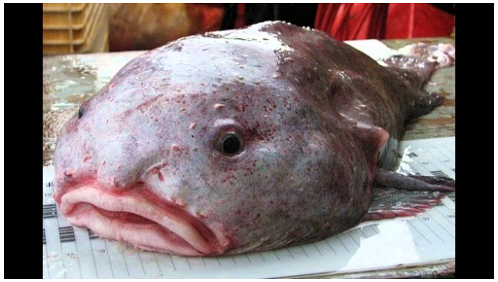
$\exists x.\ (Blobfish(x) \land Cute(x))$



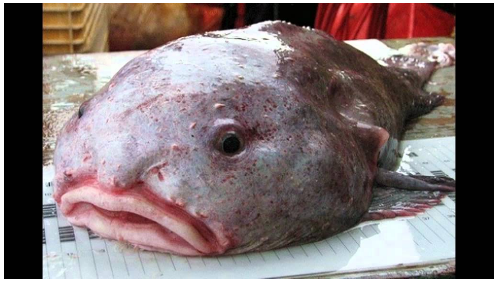
A Correct Translation

Some blobfish is cute.

$\exists x. \; (\mathit{Blobfish}(x) \land \cancel{\mathit{Cute}(x)})$



A Correct Translation



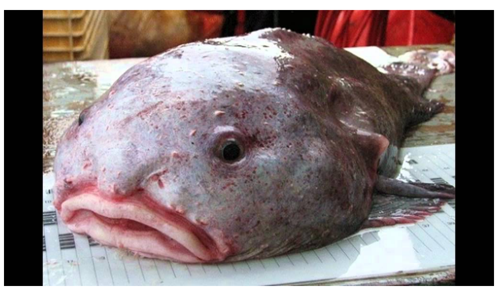
Some blobfish is cute.

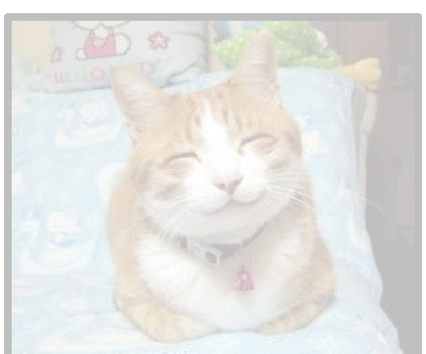
$\exists x.\ (\cancel{\mathit{Blobfish}(x) \land \mathit{Cute}(x)})$

A Correct Translation

Some blobfish is cute.

$\exists x. \, (\mathit{Blobfish}(x) \land \mathit{Cute}(x))$





A Correct Translation

Some blobfish is cute.

$\exists x.\ (\mathit{Blobfish}(x) \land \mathit{Cute}(x))$

A Correct Translation

Some blobfish is cute.

$\exists x.\ (\cancel{Blobfish(x)} \land Cute(x))$

A Correct Translation

Some blobfish is cute.

$$\exists x. (\cancel{\mathit{Blobfish}(x) \land \mathit{Cute}(x)})$$

A Correct Translation

Some blobfish is cute.

$$\exists x. \, (\mathit{Blobfish}(x) \land \mathit{Cute}(x))$$

A Correct Translation

Some blobfish is cute.

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

A Correct Translation

Some blobfish is cute.

$\exists x.\ (\mathit{Blobfish}(x) \land \mathit{Cute}(x))$

A Correct Translation

Some blobfish is cute.

$\exists x.\ (\mathit{Blobfish}(x) \land \cancel{\mathit{Cute}(x)})$

A Correct Translation

Some blobfish is cute.

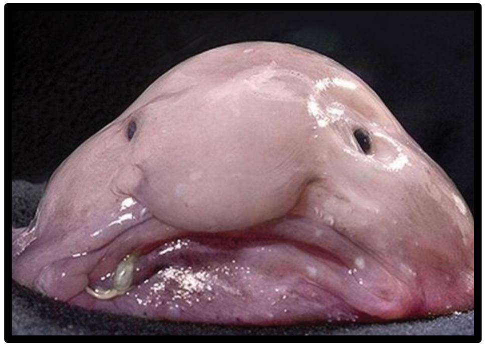
$\exists x.\ (\cancel{\mathit{Blobfish}(x)\ \land\ \mathit{Cute}(x)})$

A Correct Translation

Some blobfish is cute.

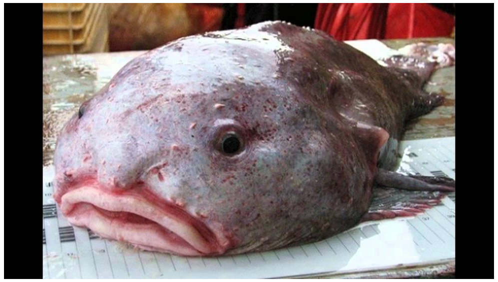
$\exists x. \, (Blobfish(x) \land Cute(x))$

A Correct Translation



Some blobfish is cute.

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



"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 $\rightarrow$.
- The $\exists$ quantifier *usually* is paired with $\wedge$.
- 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 $\wedge$ connective prevents the statement from being *true* when speaking about some object you don't care about.