

First-Order Logic

Part Two

Recap from Last Time

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 many objects at once.

Some bear is curious.

$\exists b. (Bear(b) \wedge Curious(b))$

\exists is the **existential quantifier** and says "there is a choice of b where the following is true."

“For any natural number n ,
 n is even if and only if n^2 is even”

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

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

“Some P is a Q ”

translates as

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

Useful Intuition:

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

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

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

“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 .

New Stuff!

The Aristotelian Forms

“All As are Bs”

$$\forall x. (A(x) \rightarrow B(x))$$

“Some As are Bs”

$$\exists x. (A(x) \wedge B(x))$$

“No As are Bs”

$$\forall x. (A(x) \rightarrow \neg B(x))$$

“Some As aren't Bs”

$$\exists x. (A(x) \wedge \neg B(x))$$

It is worth committing these patterns to memory. We'll be using them throughout the day and they form the backbone of many first-order logic translations.

The Art of Translation

Using the predicates

- *Person*(p), which states that p is a person, and
- *Loves*(x, y), which states that x loves y ,

write a sentence in first-order logic that means “every person loves someone else.”

Answer at

<https://cs103.stanford.edu/pollev>

$$\forall p. (Person(p) \rightarrow$$
$$\quad \exists q. (Person(q) \wedge p \neq q \wedge$$
$$\quad \quad Loves(p, q)$$
$$\quad)$$
$$)$$

Using the predicates

- *Person*(p), which states that p is a person, and
- *Loves*(x, y), which states that x loves y ,

write a sentence in first-order logic that means “there is a person that everyone else loves.”

Answer at

<https://cs103.stanford.edu/pollev>

$$\begin{aligned} & \exists p. (Person(p) \wedge \\ & \quad \forall q. (Person(q) \wedge p \neq q \rightarrow \\ & \quad \quad Loves(q, p) \\ & \quad) \\ &) \end{aligned}$$

Quantifier Ordering

Combining Quantifiers

- Most interesting statements in first-order logic require a combination of quantifiers.
- Example: “Every person loves someone else”

For every person... $\forall p. (Person(p) \rightarrow$
... there is another person ... $\exists q. (Person(q) \wedge p \neq q \wedge$
... they love $Loves(p, q)$
)
)

Combining Quantifiers

- Most interesting statements in first-order logic require a combination of quantifiers.
- Example: “There is someone everyone else loves.”

There is a person... $\exists p. (Person(p) \wedge$
... that everyone else ... $\forall q. (Person(q) \wedge p \neq q \rightarrow$
... loves. $Loves(q, p)$
)
)

For Comparison

For every person... $\forall p. (Person(p) \rightarrow$

... there is another person ... $\exists q. (Person(q) \wedge p \neq q \wedge$
... they love $Loves(p, q)$

)
)

There is a person... $\exists p. (Person(p) \wedge$

... that everyone else ... $\forall q. (Person(q) \wedge p \neq q \rightarrow$
... loves. $Loves(q, p)$

)
)

Quantifier Ordering

- Consider these two first-order formulas:

$$\forall m. \exists n. m < n.$$

$$\exists n. \forall m. m < n.$$

- Pretend for the moment that our world consists purely of natural numbers, so the variables m and n refer specifically to natural numbers.
- One of these statements is true. The other is false.
- Which is which?
- Why?

Answer at

<https://cs103.stanford.edu/pollev>

Quantifier Ordering

- Consider these two first-order formulas:

$$\forall m. \exists n. m < n.$$

$$\exists n. \forall m. m < n.$$

- This says

**for every natural number m ,
there's a larger natural number n .**

- This is true: given any $m \in \mathbb{N}$, we can choose n to be $m + 1$.
- Notice that we can pick n based on m , and we don't have to pick the same n each time.

Quantifier Ordering

- Consider these two first-order formulas:

$$\forall m. \exists n. m < n.$$

$$\exists n. \forall m. m < n.$$

- This says

**there is a natural number n
that's larger than every natural number m**

- This is false: no natural number is bigger than every natural number.
- Because $\exists n$ comes first, we have to make a single choice of n that works regardless of what we choose for m .

Quantifier Ordering

- The statement

$$\forall x. \exists y. P(x, y)$$

means “for any choice of x , there's some choice of y where $P(x, y)$ is true.”

- The choice of y can be different every time and can depend on x .

Quantifier Ordering

- The statement

$$\exists x. \forall y. P(x, y)$$

means “there is some x where for any choice of y , we get that $P(x, y)$ is true.”

- Since the inner part has to work for any choice of y , this places a lot of constraints on what x can be.

Order matters when mixing existential
and universal quantifiers!

Time-Out for Announcements!

Problem Set Two

- Problem Set One was due today at 1:00PM.
 - You can extend the deadline to 1:00PM Saturday using one of your late days. As usual, no late submissions will be accepted beyond 1:00PM Saturday without prior approval.
- Problem Set Two goes out today. It's due next Friday at 1:00PM.
 - Explore first-order logic!
 - Expand your proofwriting toolkit!
- We have some online readings for this problem set.
 - Check out the ***Guide to Logic Translations*** for more on how to convert from English to FOL.
 - Check out the ***Guide to Negations*** for information about how to negate formulas.
 - Check out the ***First-Order Translation Checklist*** for details on how to check your work.

A Music Recommendation



Back to CS103!

Mechanics: Negating Statements

An Extremely Important Table

When is this true?

When is this false?

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

For all objects x ,
 $P(x)$ is true.

$$\exists x. \neg P(x)$$

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

There is an x where
 $P(x)$ is true.

$$\forall x. \neg P(x)$$

$$\forall x. \neg P(x)$$

For all objects x ,
 $P(x)$ is false.

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

$$\exists x. \neg P(x)$$

There is an x where
 $P(x)$ is false.

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

Negating First-Order Statements

- Use the equivalences

$\neg \forall x. A$ is equivalent to $\exists x. \neg A$

$\neg \exists x. A$ is equivalent to $\forall x. \neg A$

to negate quantifiers.

- Mechanically:
 - Push the negation across the quantifier.
 - Change the quantifier from \forall to \exists or vice-versa.
- Use techniques from propositional logic to negate connectives.

Taking a Negation

$\forall x. \exists y. \text{Loves}(x, y)$
(“Everyone loves someone.”)

$\neg \forall x. \exists y. \text{Loves}(x, y)$

$\exists x. \neg \exists y. \text{Loves}(x, y)$

$\exists x. \forall y. \neg \text{Loves}(x, y)$

(“There's someone who doesn't love anyone.”)

Two Useful Equivalences

- The following equivalences are useful when negating statements in first-order logic:

$\neg(p \wedge q)$ is equivalent to $p \rightarrow \neg q$

$\neg(p \rightarrow q)$ is equivalent to $p \wedge \neg q$

- These identities are useful when negating statements involving quantifiers.
 - \wedge is used in existentially-quantified statements.
 - \rightarrow is used in universally-quantified statements.
- When pushing negations across quantifiers, we *strongly recommend* using the above equivalences to keep \rightarrow with \forall and \wedge with \exists .

Negating Quantifiers

- What is the negation of the following statement, which says “there is a cute puppy”?

$$\exists x. (\mathit{Puppy}(x) \wedge \mathit{Cute}(x))$$

- We can obtain it as follows:

$$\neg \exists x. (\mathit{Puppy}(x) \wedge \mathit{Cute}(x))$$

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

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

- This says “no puppy is cute.”
- Do you see why this is the negation of the original statement from both an intuitive and formal perspective?

$\exists S. (Set(S) \wedge \forall x. x \notin S)$
(“There is a set with no elements.”)

$\neg \exists S. (Set(S) \wedge \forall x. x \notin S)$
 $\forall S. \neg (Set(S) \wedge \forall x. \neg x \notin S)$
 $\forall S. (Set(S) \rightarrow \neg \forall x. x \notin S)$
 $\forall S. (Set(S) \rightarrow \exists x. \neg (x \notin S))$
 $\forall S. (Set(S) \rightarrow \exists x. x \in S)$

(“Every set contains at least one element.”)

Restricted Quantifiers

Quantifying Over Sets

- The notation

$$\forall x \in S. P(x)$$

means “for any element x of set S , $P(x)$ holds.” (It’s vacuously true if S is empty.)

- The notation

$$\exists x \in S. P(x)$$

means “there is an element x of set S where $P(x)$ holds.” (It’s false if S is empty.)

Quantifying Over Sets

- The syntax

$$\forall x \in S. P(x)$$

$$\exists x \in S. P(x)$$

is allowed for quantifying over sets.

- In CS103, feel free to use these restricted quantifiers, but please do not use variants of this syntax.
- For example, don't do things like this:

$$\forall x \text{ with } P(x). Q(x)$$

$$\forall y \text{ such that } P(y) \wedge Q(y). R(y).$$

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

Expressing Uniqueness

Using the predicate

- *WayToFindOut*(w), which states that w is a way to find out,

write a sentence in first-order logic that means “there is only one way to find out.”

$\exists w. (WayToFindOut(w) \wedge$
 $\forall x. (WayToFindOut(x) \rightarrow x = w)$
)

Expressing Uniqueness

- To express the idea that there is exactly one object with some property, we write that
 - there exists at least one object with that property, and that
 - there are no other objects with that property.
- You sometimes see a special “uniqueness quantifier” used to express this:

$$\exists!x. P(x)$$

- For the purposes of CS103, please do not use this quantifier. We want to give you more practice using the regular \forall and \exists quantifiers.

Next Time

- ***Functions***
 - How do we model transformations and pairings?
- ***First-Order Definitions***
 - Where does first-order logic come into all of this?
- ***Proofs with Definitions***
 - How does first-order logic interact with proofs?