Mathematical Logic

Part Three
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 muggle is intelligent.

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

$\exists$ is the **existential quantifier** and says "for some choice of $m$, the following is true."
∀n. (n ∈ ℕ → (Even(n) ↔ Even(n^2)))

∀ is the universal quantifier and says "for any choice of n, the following is true."
“All A's are B's” translates as

∀x. (A(x) → B(x))
Useful Intuition:

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

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

If \( x \) is a counterexample, it must have property \( A \) but not have property \( B \).
“Some $A$ is a $B$” translates as

$\exists x. (A(x) \land B(x))$
Useful Intuition:

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

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

If $x$ is an example, it must have property $A$ on top of property $B$. 
The Aristotelian Forms

“All As are Bs”
\[ \forall x. (A(x) \rightarrow B(x)) \]

“Some As are Bs”
\[ \exists x. (A(x) \land B(x)) \]

“No As are Bs”
\[ \forall x. (A(x) \rightarrow \neg B(x)) \]

“Some As aren’t Bs”
\[ \exists x. (A(x) \land \neg B(x)) \]

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

- \textit{Person}(p), which states that \( p \) is a person, and
- \textit{Loves}(x, y), which states that \( x \) loves \( y \),

write a sentence in first-order logic that means “everybody loves someone else.”
∀p. (Person(p) →
  ∃q. (Person(q) ∧ p ≠ q ∧
  Loves(p, q))
)
How many of the following first-order logic statements are correct translations of “everyone loves someone else?”

<table>
<thead>
<tr>
<th>Statement</th>
<th>Correct</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\forall p. (\text{Person}(p) \rightarrow \exists q. (\text{Person}(q) \land Loves(p, q))$</td>
<td></td>
</tr>
<tr>
<td>$\forall p. (\text{Person}(p) \land \exists q. (\text{Person}(q) \land p \neq q \land Loves(p, q))$</td>
<td></td>
</tr>
<tr>
<td>$\forall p. (\text{Person}(p) \rightarrow \exists q. (\text{Person}(q) \land p \neq q \rightarrow Loves(p, q))$</td>
<td></td>
</tr>
<tr>
<td>$\exists p. (\text{Person}(p) \rightarrow \forall q. (\text{Person}(q) \land p \neq q \land Loves(p, q))$</td>
<td></td>
</tr>
</tbody>
</table>

Answer at PollEv.com/cs103 or text CS103 to 22333 once to join, then 0, 1, 2, 3, or 4.
Using the predicates

- \textit{Person}(p), which states that \( p \) is a person, and
- \textit{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.”
\[ \exists p. (Person(p) \land \\
\forall q. (Person(q) \land p \neq q \rightarrow Loves(q, p)) \]
Combining Quantifiers

• Most interesting statements in first-order logic require a combination of quantifiers.

• Example: “Everyone loves someone else.”

\[ \forall p. (\text{Person}(p) \rightarrow \exists q. (\text{Person}(q) \land p \neq q \land \text{Loves}(p, q))) \]

For every person, there is some person who isn’t them that they love.
Combining Quantifiers

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

$$\exists p. \ (\text{Person}(p) \land \forall q. \ (\text{Person}(q) \land p \neq q \rightarrow \text{Loves}(q, p)))$$

- There is some person who everyone who isn’t them loves.
For every person,
there is some person who isn't them that they love.

There is some person who everyone who isn't them loves.
Everyone Loves Someone Else

No one here is universally loved.
There is Someone Everyone Else Loves

This person does not love anyone else.

This person does not love anyone else.
Everyone Loves Someone Else *and*
There is Someone Everyone Else Loves
\[ \forall p. \ (\text{Person}(p) \rightarrow \exists q. \ (\text{Person}(q) \land p \neq q \land \text{Loves}(p, q))) \]

For every person, there is some person who isn't them that they love.

\[ \exists p. \ (\text{Person}(p) \land \forall q. \ (\text{Person}(q) \land p \neq q \rightarrow \text{Loves}(q, p))) \]

There is some person who everyone who isn't them loves.
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!
Which person in this diagram do you most aspire to be?

Answer at PollEv.com/cs103 or text CS103 to 22333 once to join, then A, B, C, or D.
Time-Out for Announcements!
Problem Set Two

• Problem Set Two is due this Friday at 2:30PM.
  • Once we’re done with this lecture, you’ll know everything you need to complete it!
  • Have questions? Feel free to stop by office hours or to ask on Piazza.
• Hopefully you’ve taken a few minutes to read over all the problems by now. If not, we’d strongly recommend doing so.
• **Good idea:** Aim to complete Q1 – Q5 by the end of the evening.
Problem Set One Solutions

- Problem Set One solutions are now available.
- **Please take the time to read over these solutions.**
  - For non-proof questions, make sure that you understand the intuition behind the answers. If they match yours, great! If not, that would be a great question to ask us.
  - For proofs, look over the style and formatting. Compare them against yours. How do they compare?
  - Each question has a “Why We Asked This Question” section at the end. Make sure you read over it – it would be a shame if you did a problem and didn’t hit the key insight we wanted you to have.
Apply to Section Lead!

- Want to teach a CS106A/B/X section? Already completed CS106B or CS106X? Apply to section lead at https://cs198.stanford.edu
- Application is due Thursday, February 1st.
- There’s a second round of hiring later this quarter for folks currently in CS106B/X – stay tuned!
- This is an amazing program. Highly recommended!
Back to CS103!
Set Translations
Using the predicates

- \( Set(S) \), which states that \( S \) is a set, and
- \( x \in y \), which states that \( x \) is an element of \( y \),

write a sentence in first-order logic that means “the empty set exists.”
Both of these translations are correct. Just like in propositional logic, there are many different equivalent ways of expressing the same statement in first-order logic.

\[ \exists S. (\text{Set}(S) \land \neg \exists x. x \in S) \]

\[ \exists S. (\text{Set}(S) \land \forall x. x \notin S) \]
Using the predicates

- $Set(S)$, which states that $S$ is a set, and
- $x \in y$, which states that $x$ is an element of $y$,

write a sentence in first-order logic that means “two sets are equal if and only if they contain the same elements.”
\(\forall S. (Set(S) \rightarrow \forall T. (Set(T) \rightarrow (S = T \leftrightarrow \forall x. (x \in S \leftrightarrow x \in T)))\)
∀S. (Set(S) →
   ∀T. (Set(T) →
       (S = T ↔ ∀x. (x ∈ S ↔ x ∈ T)))
 )
)
Mechanics: Negating Statements
Which of the following is the negation of the statement \( \forall x. \exists y. Loves(x, y) \)?

A. \( \forall x. \forall y. \neg Loves(x, y) \)
B. \( \forall x. \exists y. \neg Loves(x, y) \)
C. \( \exists x. \forall y. \neg Loves(x, y) \)
D. \( \exists x. \exists y. \neg Loves(x, y) \)
E. None of these.
F. Two or more of these.

Answer at PollEv.com/cs103 or text CS103 to 22333 once to join, then A, B, C, D, E, or F.
<table>
<thead>
<tr>
<th></th>
<th>When is this true?</th>
<th>When is this false?</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\forall x. P(x)$</td>
<td>For any choice of $x$, $P(x)$</td>
<td>$\exists x. \neg P(x)$</td>
</tr>
<tr>
<td>$\exists x. P(x)$</td>
<td>For some choice of $x$, $P(x)$</td>
<td>$\forall x. \neg P(x)$</td>
</tr>
<tr>
<td>$\forall x. \neg P(x)$</td>
<td>For any choice of $x$, $\neg P(x)$</td>
<td>$\exists x. P(x)$</td>
</tr>
<tr>
<td>$\exists x. \neg P(x)$</td>
<td>For some choice of $x$, $\neg P(x)$</td>
<td>$\forall x. P(x)$</td>
</tr>
</tbody>
</table>
Negating First-Order Statements

• Use the equivalences

$$\neg \forall x. A \equiv \exists x. \neg A$$

$$\neg \exists x. A \equiv \forall x. \neg A$$

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

∀x. ∃y. Loves(x, y)  
(“Everyone loves someone.”)

¬∀x. ∃y. Loves(x, y)  
∃x. ¬∃y. Loves(x, y)  
∃x. ∀y. ¬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:
  \[ \neg(p \land q) \equiv p \rightarrow \neg q \]
  \[ \neg(p \rightarrow q) \equiv p \land \neg q \]

- These identities are useful when negating statements involving quantifiers.
  - $\land$ 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 $\land$ with $\exists$. 
Negating Quantifiers

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

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

• We can obtain it as follows:

$$\neg \exists x. (Puppy(x) \land Cute(x))$$
$$\forall x. \neg(Puppy(x) \land Cute(x))$$
$$\forall x. (Puppy(x) \rightarrow \neg Cute(x))$$

• 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?
\[\exists S. \ (\text{Set}(S) \land \forall x. \ \neg(x \in S))\]
(“There is a set with no elements.”)

\[\neg \exists S. \ (\text{Set}(S) \land \forall x. \ \neg(x \in S))\]
\[\forall S. \ \neg(\text{Set}(S) \land \forall x. \ \neg(x \in S))\]
\[\forall S. \ (\text{Set}(S) \rightarrow \neg \forall x. \ \neg(x \in S))\]
\[\forall S. \ (\text{Set}(S) \rightarrow \exists x. \ \neg \neg(x \in S))\]
\[\forall S. \ (\text{Set}(S) \rightarrow \exists x. \ x \in S)\]
(“Every set contains at least one element.”)
These two statements are not negations of one another. Can you explain why?

$$\exists S. (\text{Set}(S) \land \forall x. \neg(x \in S))$$

(“There is a set that doesn't contain anything”)

$$\forall S. (\text{Set}(S) \land \exists x. (x \in S))$$

(“Everything is a set that contains something”)

Remember: $\forall$ usually goes with $\rightarrow$, not $\land$.
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

\[ \forall x \in S. \varphi \]
\[ \exists x \in S. \varphi \]

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:

⚠ \[ \forall x \text{ with } P(x). \ Q(x) \]
⚠ \[ \forall y \text{ such that } P(y) \land Q(y). \ R(y) \]
⚠ \[ \exists P(x). \ Q(x) \]
Expressing Uniqueness
Using the predicate

- \( \text{Level}(l) \), which states that \( l \) is a level,

write a sentence in first-order logic that means “there is only one level.”

A fun diversion:

http://www.onemorelevel.com/game/there_is_only_one_level
$\exists l. \ (\text{Level}(l) \land \forall x. \ (x \neq l \to \neg\text{Level}(x)))$
\[ \exists l. (\text{Level}(l) \land \\
\forall x. (\text{Level}(x) \rightarrow x = l) ) \]
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

- **Binary Relations**
  - How do we model connections between objects?
- **Equivalence Relations**
  - How do we model the idea that objects can be grouped into clusters?
- **First-Order Definitions**
  - Where does first-order logic come into all of this?
- **Proofs with Definitions**
  - How does first-order logic interact with proofs?