Mathematical Logic
Where We Are Now

- Our coverage of logic focused on these topics:
  - Propositional variables.
  - Propositional connectives.
  - Propositional equivalences.
  - Predicates, functions, and constant symbols.
  - Objects and propositions.
  - Quantifiers.
  - Evaluating first-order formulas relative to a world.
  - Translating into first-order logic.
  - Negating and simplifying first-order formulas.
- Your goal this week is to keep your proof skills sharp while mastering the ins and outs of first-order logic.
Things You Should Do Today

- Review the solutions for Problem Set One and make sure you *completely* and *unambiguously* understand the answers. Ask for help if this isn't the case!
- Read the “Guide to Negating Formulas” and “Guide to First-Order Logic Translations” on the course website to get more exposure and practice with those skills.
- Continue working through PS2.
Things You Should Do Tomorrow

- Look over your feedback on PS1 (and the PS2 checkpoint) and make sure you understand all the feedback you get completely and unambiguously. Ask the course staff for help, either on Piazza or in office hours, if you don't.

- Continue working on PS2.

- Start reviewing your partner's answers and compiling a single, definitive set of answers that you're going to turn in.

- Stop by office hours to get feedback on your proofs and take that feedback seriously.

- (Also, complete the **CS103A assignment** after Wednesday’s lecture—it is due Friday at 2:30 pm!)
Topics You Wanted More Practice With

- First-order logic and translations [today’s topic]
- Truth tables
- Proofs and disproofs [more next week]
  - For now, make sure you look at the PS1 solutions and feedback!!
Mechanics: *Negating Statements*
∀p. (Person(p) →
   ∃q. (Person(q) ∧ p ≠ q ∧
       Loves(p, q))
)
)
¬∀p. (Person(p) →
    ∃q. (Person(q) ∧ p ≠ q ∧
        Loves(p, q)
    )
)
\[ \neg \forall p. \ (Person(p) \rightarrow \exists q. \ (Person(q) \land p \neq q \land Loves(p, q)) \]
\[
\neg \forall p. \ (\text{Person}(p) \rightarrow \\
\exists q. \ (\text{Person}(q) \land p \neq q \land \\
\text{Loves}(p, q)) \\
\) \\
\) \\
\)
\]

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

\hline
\neg \forall x. A
\hline
\exists x. \neg A
\[ \neg \forall p. (\text{Person}(p) \rightarrow \\
\quad \exists q. (\text{Person}(q) \land p \neq q \land \\
\quad \text{Loves}(p, q) \\
\quad ) \\
\quad ) \]

\[ \begin{array}{c}
\neg \forall x. A \\
\hline
\exists x. \neg A
\end{array} \]
\[\exists p. \neg (\text{Person}(p) \rightarrow \exists q. (\text{Person}(q) \land p \neq q \land \text{Loves}(p, q)) \) \]
\[ \exists p. \neg(Person(p) \rightarrow \exists q. (Person(q) \land p \neq q \land Loves(p, q)) \) \) \)
\( \exists p. \neg(Person(p) \rightarrow
\exists q. (Person(q) \land p \neq q \land Loves(p, q))\)
∀p. ¬(Person(p) →
   ∃q. (Person(q) ∧ p ≠ q ∧ Loves(p, q)
   )
)
\[\neg(A \rightarrow B)\]
\[\frac{}{A \land \neg B}\]
\[ \exists p. \neg (\text{Person}(p) \rightarrow \\
\exists q. (\text{Person}(q) \land p \neq q \land \\
\text{Loves}(p, q)) \]

\[ \neg (A \rightarrow B) \]

\[ A \land \neg B \]
∃p. (Person(p) \land 
\neg \exists q. (Person(q) \land p \neq q \land Loves(p, q))
)

\neg(A \rightarrow B)

A \land \neg B
\( \exists p. \ (Person(p) \land \neg \exists q. \ (Person(q) \land p \neq q \land Loves(p, q)) \) \)
∃p. (Person(p) \land
  \neg\exists q. (Person(q) \land p \neq q \land
  Loves(p, q))
)

\neg\exists x. A

\forall x. \neg A
\( \exists p. (\text{Person}(p) \land \\
\neg \exists q. (\text{Person}(q) \land p \neq q \land \\
\text{Loves}(p, q)) \\
\) \\
\) \\
\)

\[ \neg \exists x. A \\
\hline \\
\forall x. \neg A \]
\[ \exists p. (\text{Person}(p) \land \\
\neg \exists q. (\text{Person}(q) \land p \neq q \land \\
\text{Loves}(p, q)) \\
\) \]

\[
\begin{prooftree}
\neg \exists x. A \\
\hline
\forall x. \neg A
\end{prooftree}
\]
\[ \exists p. \ (\text{Person}(p) \land \forall q. \ \neg (\text{Person}(q) \land p \neq q \land \text{Loves}(p, q)) \) \]
\[ \exists p. \ (\text{Person}(p) \land \\
\forall q. \ \neg (\text{Person}(q) \land p \neq q \land \\
\quad \text{Loves}(p, q)) \]
\[ \exists p. (\text{Person}(p) \land \forall q. \neg(\text{Person}(q) \land p \neq q \land \text{Loves}(p, q)) \) \]

\[ \neg(A \land B) \quad \overline{\quad A \rightarrow \neg B \quad} \]
\[ \exists p. (\text{Person}(p) \land \\
\forall q. \neg(\text{Person}(q) \land p \neq q \land \\
\text{Loves}(p, q)) \]

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

\[
\neg (A \land B) \\
\hline
A \rightarrow \neg B
\]
∃p. (Person(p) ∧
   ∀q. (Person(q) ∧ p ≠ q →
       ¬Loves(p, q)
   )
)
∀p. (Person(p) →
   ∃q. (Person(q) ∧ p ≠ q ∧
       Loves(p, q)
   )
)

∃p. (Person(p) ∧
   ∀q. (Person(q) ∧ p ≠ q →
       ¬Loves(p, q)
   )
)
∃p. (Person(p) ∧
   ∀q. (Person(q) ∧ p ≠ q → Loves(q, p))
)

Your turn!

Try negating this formula with
the other folks at your table.
See what you come up with!
\( \neg \exists p. \ (\text{Person}(p) \land \forall q. \ (\text{Person}(q) \land p \neq q \rightarrow \text{Loves}(q, p)) \) \)
∀p. ¬(Person(p) ∧
    ∀q. (Person(q) ∧ p ≠ q → Loves(q, p)
)
)
∀p. (Person(p) → 
  ¬∀q. (Person(q) ∧ p ≠ q → Loves(q, p)) 
)
∀p. (Person(p) →
   ∃q. ¬(Person(q) ∧ p ≠ q →
       Loves(q, p))
)
)
∀p. (Person(p) →
  ∃q. (Person(q) ∧ p ≠ q ∧
    ¬Loves(q, p)
  )
)
\[ \exists p. (\text{Person}(p) \land \\
\quad \forall q. (\text{Person}(q) \land p \neq q \rightarrow \\
\quad \quad \text{Loves}(q, p) \\
\quad ) \\
) \]

\[ \forall p. (\text{Person}(p) \rightarrow \\
\quad \exists q. (\text{Person}(q) \land p \neq q \land \\
\quad \quad \neg\text{Loves}(q, p) \\
\quad ) \\
) \]
Techniques: *Translating Statements*
Common Patterns

- A statement of the form
  \[ \forall x. (P(x) \rightarrow Q(x)) \]
  can be read as “all P's are Q's.”

- A statement of the form
  \[ \exists x. (P(x) \land Q(x)) \]
  can be read as “there is a P that is also a Q” or “some P's are Q's.”

- **Remember:** If you see \( \exists \) paired with \( \rightarrow \) or \( \forall \) paired with \( \land \), the statement is probably incorrect!
Given the predicates

- \textit{Person}(p), which states that \( p \) is a person, and
- \textit{CanLearnFrom}(x, y), which says that \( x \) can learn from \( y \),

write a statement in first-order logic that says “everyone has someone they can learn from.”
Everyone has someone they can learn from
Every person $p$ has someone they can learn from
Every person $p$ has someone they can learn from

“All As are Bs.”

$\forall x. (A(x) \rightarrow B(x))$
∀p. (Person(p) → 
   p has someone they can learn from 

)
∀p. (Person(p) →
   there is a person q that p can learn from
 )
∀p. (Person(p) →
    there is a person q that p can learn from
)

“Some As are Bs.”
∃x. (A(x) ∧ B(x))
∀p. (Person(p) →
    ∃q. (Person(q) ∧
        p can learn from q
    )
)

∀p. (Person(p) →
    ∃q. (Person(q) ∧
        CanLearnFrom(p, q)
    )
)
∀p. (Person(p) →
  ∃q. (Person(q) ∧
       CanLearnFrom(p, q))
  )
  )

☐ Pair quantifiers with the appropriate connectives.
☐ Use whitespace and indentation to clarify meaning.
☐ Defer quantifiers until they’re needed.
☐ Check the types of all terms in the formula.
☐ Check the scoping on each variable.
Consider this statement:

“If someone is happy, then everyone is happy.”

What is the *contrapositive* of this statement?
If someone is happy, then everyone is happy
someone is happy $\rightarrow$ everyone is happy
someone is happy \rightarrow (\forall x. \ Happy(x))
(∃x. Happy(x)) → (∀x. Happy(x))
\((\exists x. \text{Happy}(x)) \rightarrow (\forall x. \text{Happy}(x))\)
\((\exists x. \text{Happy}(x)) \rightarrow (\forall x. \text{Happy}(x))\)
\neg (\forall x. \text{Happy}(x)) \rightarrow \neg (\exists x. \text{Happy}(x))
$(\exists x. \neg \text{Happy}(x)) \rightarrow \neg (\exists x. \text{Happy}(x))$
\((\exists x. \neg \text{Happy}(x)) \rightarrow (\forall x. \neg \text{Happy}(x))\)
CONTRAPOSITIVE of “If someone is happy, then everyone is happy”

\[(\exists x. \neg \text{Happy}(x)) \rightarrow (\forall x. \neg \text{Happy}(x))\]

“If someone is not happy, then everyone is not happy.”
Consider this statement:

“If someone is happy, then everyone is happy.”

What is the *negation* of this statement?
(\exists x. \text{Happy}(x)) \rightarrow (\forall x. \text{Happy}(x))
(∃x. Happy(x)) → (∀x. Happy(x))

¬((∃x. Happy(x)) → (∀x. Happy(x)))
(∃x. Happy(x)) → (∀x. Happy(x))

(∃x. Happy(x)) ∧ ¬(∀x. Happy(x))
\[(\exists x. \text{Happy}(x)) \rightarrow (\forall x. \text{Happy}(x))\]

\[(\exists x. \text{Happy}(x)) \land (\exists x. \neg \text{Happy}(x))\]
"If someone is happy, then everyone is happy"

\[(\exists x. \text{Happy}(x)) \rightarrow (\forall x. \text{Happy}(x))\]

\[(\exists x. \text{Happy}(x)) \land (\exists x. \neg \text{Happy}(x))\]

“Someone is happy and someone is not happy.”