Mathematical Logic
Where We Are Now

• Week 2 covered these key 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.
Where We're Going

• Week 3 is about *discrete structures*:
  • Binary relations. (Yesterday)
  • Equivalence relations. (Yesterday/Tomorrow)
  • Strict order relations (Tomorrow)
  • Functions (Friday)

• From yesterday, you should know what a binary relation is, what the terms *reflexive*, *symmetric*, and *transitive* mean, and how to prove that a binary relation has those properties.
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 Translation” 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 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. If at all possible, aim to complete two of Q8, Q9, and Q10 and start writing up your answers.
- 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.
Mechanics: *Negating Statements*
∀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 \\
\qquad Loves(p, q))}
∀p. (Person(p) → 
    ∃q. (Person(q) ∧ p ≠ q ∧ Loves(p, q))
)
\( \neg \forall p. (\text{Person}(p) \rightarrow \\
\exists q. (\text{Person}(q) \land p \neq q \land \\
\text{Loves}(p, q)) \\
) \\
) \)
\( \neg \forall p. (\text{Person}(p) \rightarrow \exists q. (\text{Person}(q) \land p \neq q \land \text{Loves}(p, q)) \) 
\)

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

\begin{array}{c}
\neg \forall x. A \\
\hline
\exists x. \neg A
\end{array}
\( \exists p. \neg (\text{Person}(p) \rightarrow \exists q. (\text{Person}(q) \land p \neq q \land \text{Loves}(p, q)) \) )

\( \neg \forall x. A \quad \underline{\quad \quad \quad \quad \quad \quad \quad} \quad \exists x. \neg A \)
\[ \exists p. \neg (\text{Person}(p) \rightarrow \exists q. (\text{Person}(q) \land p \neq q \land \text{Loves}(p, q)) ) \]
∃p. ¬(Person(p) →
  ∃q. (Person(q) ∧ p ≠ q ∧
       Loves(p, q))
  )
)

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

\neg (A \rightarrow B) \quad \therefore A \land \neg B
\[ \exists p. \neg (\text{Person}(p) \rightarrow \exists q. (\text{Person}(q) \land p \neq q \land \text{Loves}(p, q)) ) \]
\[ \exists p. \ (\text{Person}(p) \land \neg \exists q. \ (\text{Person}(q) \land p \neq q \land \text{Loves}(p, q)) \land \neg (A \rightarrow B) \) \]
\[ \exists p. (\text{Person}(p) \land \\
\neg \exists q. (\text{Person}(q) \land p \neq q \land \\
\quad \text{Loves}(p, q) \\
\) ) \) \\
\]
\[ \exists p. \ (\text{Person}(p) \land \\
\neg \exists q. \ (\text{Person}(q) \land p \neq q \land \\
\text{Loves}(p, q)) \) \]
\[ \exists p. \ (\text{Person}(p) \land \neg \exists q. \ (\text{Person}(q) \land p \neq q \land \text{Loves}(p, q)) \) \]
∃p. (Person(p) ∧ ¬∃q. (Person(q) ∧ p ≠ q ∧ Loves(p, q))

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

\[ \forall x. \neg A \]

\[ \neg \exists x. A \]

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

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

\[ \underline{\neg(A \land B)} \\
A \rightarrow \neg B \]
∃p. (Person(p) ∧ 
    ∀q. ¬(Person(q) ∧ p ≠ q ∧ Loves(p, q)
    )
)
\[ \exists p. \ (\text{Person}(p) \land \\
\forall q. \ \neg (\text{Person}(q) \land p \neq q \land \\
\text{Loves}(p, q)) \]
\[ \exists p. \ (\text{Person}(p) \land \\
\forall q. \ (\text{Person}(q) \land p \neq q \rightarrow \\
\neg \text{Loves}(p, q)) \) \]
∃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)
    )
) )
\[ \exists p. (\text{Person}(p) \land \\
\forall q. (\text{Person}(q) \land p \neq q \to \\
\text{Loves}(q, p)) \]
\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 \\
\forall q. \ (\text{Person}(q) \land p \neq q \rightarrow \\
\quad \text{Loves}(q, p)) \\
\) \\
\]

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

- \( \text{Person}(p) \), which states that \( p \) is a person, and
- \( \text{CanLearnFrom}(x, y) \), 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
)
\forall p. (\text{Person}(p) \rightarrow \\
\text{there is a person } q \text{ that } p \text{ 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))
  )
)
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 → (∀x. Happy(x))
(∃x. Happy(x)) → (∀x. Happy(x))
(∃x. \text{Happy}(x)) \rightarrow (∀x. \text{Happy}(x))
(∃x. Happy(x)) → (∀x. Happy(x))
\neg (\forall x. \text{Happy}(x)) \rightarrow \neg (\exists x. \text{Happy}(x))
\((\exists x. \neg \text{Happy}(x)) \rightarrow \neg (\exists x. \text{Happy}(x))\)
\( (\exists x. \neg \text{Happy}(x)) \rightarrow (\forall x. \neg \text{Happy}(x)) \)
(∃x. ¬Happy(x)) → (∀x. ¬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?
\[(\exists x. \text{Happy}(x)) \rightarrow (\forall x. \text{Happy}(x))\]
\[(\exists x. \text{Happy}(x)) \rightarrow (\forall x. \text{Happy}(x))\]

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

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

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

(∃x. Happy(x)) ∧ (∃x. ¬Happy(x))

“Someone is happy and someone is not happy.”