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 checkpoint problem from Problem Set Two 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. Aim to complete Q1 Q6 by tonight, if possible.

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 Q7, Q8, and Q9 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*

```
\forall p. (Person(p) \rightarrow \exists q. (Person(q) \land p \neq q \land Loves(p, q)
)
```

```
¬\forall p. (Person(p) \rightarrow \exists q. (Person(q) \land p \neq q \land Loves(p, q)
)
```

```
\neg \forall p. (Person(p) \rightarrow \exists q. (Person(q) \land p \neq q \land Loves(p, q))
```

Useful Resource:

Go to cs103.stanford.edu and read the Guide to Negating Formulas.

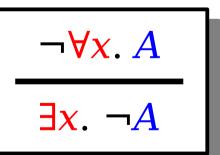
```
¬\forall p. (Person(p) \rightarrow \exists q. (Person(q) \land p \neq q \land Loves(p, q)
)
```

 $\neg \forall x. A$

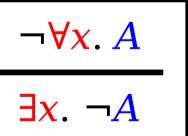
```
¬\forall p. (Person(p) \rightarrow \exists q. (Person(q) \land p \neq q \land Loves(p, q)
)
```

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

```
¬∀p. (Person(p) →
∃q. (Person(q) ∧ p ≠ q ∧
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) \\ )
```

```
\exists p. \neg (Person(p) \rightarrow \exists q. (Person(q) \land p \neq q \land Loves(p, q))
```

$$\frac{\neg (A \to B)}{A \land \neg B}$$

```
\exists p. \neg (Person(p) \rightarrow \exists q. (Person(q) \land p \neq q \land Loves(p, q))
```

$$\frac{\neg (A \to B)}{A \land \neg B}$$

```
\exists p. \neg (Person(p) \rightarrow \\ \exists q. (Person(q) \land p \neq q \land \\ Loves(p, q) \\ )
```

$$\frac{\neg (A \to B)}{A \land \neg B}$$

```
\exists p. (Person(p) \land \neg \exists q. (Person(q) \land p \neq q \land Loves(p, q))
```

$$\frac{\neg (A \to B)}{A \land \neg B}$$

```
\exists p. (Person(p) \land \neg \exists q. (Person(q) \land p \neq q \land Loves(p, q))
```

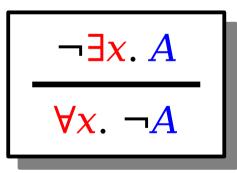
```
\exists p. (Person(p) \land \neg \exists q. (Person(q) \land p \neq q \land Loves(p, q))
```

 $\neg \exists x. A$

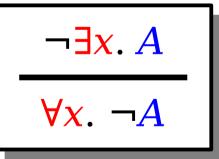
 $\forall x. \ \neg A$

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

```
\exists p. (Person(p) \land \neg \exists q. (Person(q) \land p \neq q \land Loves(p, q))
```



```
\exists p. (Person(p) \land \forall q. \neg (Person(q) \land p \neq q \land Loves(p, q))
```



```
\exists p. (Person(p) \land \forall q. \neg (Person(q) \land p \neq q \land Loves(p, q))
```

```
\exists p. (Person(p) \land \forall q. \neg (Person(q) \land p \neq q \land Loves(p, q))
```

$$\frac{\neg (A \land B)}{A \rightarrow \neg B}$$

```
\exists p. (Person(p) \land \forall q. \neg (Person(q) \land p \neq q \land Loves(p, q))
```

$$\frac{\neg (A \land B)}{A \rightarrow \neg B}$$

```
\exists p. (Person(p) \land \forall q. \neg (Person(q) \land p \neq q \land Loves(p, q))
```

$$\frac{\neg (A \land B)}{A \rightarrow \neg B}$$

```
\exists p. (Person(p) \land \forall q. (Person(q) \land p \neq q \rightarrow \neg Loves(p, q))
```

$$\frac{\neg (A \land B)}{A \rightarrow \neg B}$$

```
\exists p. (Person(p) \land \forall q. (Person(q) \land p \neq q \rightarrow \neg Loves(p, q))
```

```
\forall p. (Person(p) \rightarrow
   \exists q. (Person(q) \land p \neq q \land
       Loves(p, q)
\exists p. (Person(p) \land
   \forall q. (Person(q) \land p \neq q \rightarrow
       \neg Loves(p, q)
```

```
\exists p. (Person(p) \land \forall q. (Person(q) \land p \neq q \rightarrow Loves(q, p))
```

Your turn!

Try negating this formula with the other folks at your table. See what you come up with!

```
¬∃p. (Person(p) \land \forall q. (Person(q) \land p \neq q \rightarrow Loves(q, p))
```

```
\forall p. \neg (Person(p) \land \forall q. (Person(q) \land p \neq q \rightarrow Loves(q, p))
```

```
\forall p. (Person(p) \rightarrow \neg \forall q. (Person(q) \land p \neq q \rightarrow Loves(q, p))
```

```
\forall p. (Person(p) \rightarrow \exists q. \neg (Person(q) \land p \neq q \rightarrow Loves(q, p))
```

```
\forall p. (Person(p) \rightarrow \exists q. (Person(q) \land p \neq q \land \neg Loves(q, p))
```

```
\exists p. (Person(p) \land
   \forall q. (Person(q) \land p \neq q \rightarrow
       Loves(q, p)
\forall p. (Person(p) \rightarrow
   \exists q. (Person(q) \land p \neq q \land
       \neg 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

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

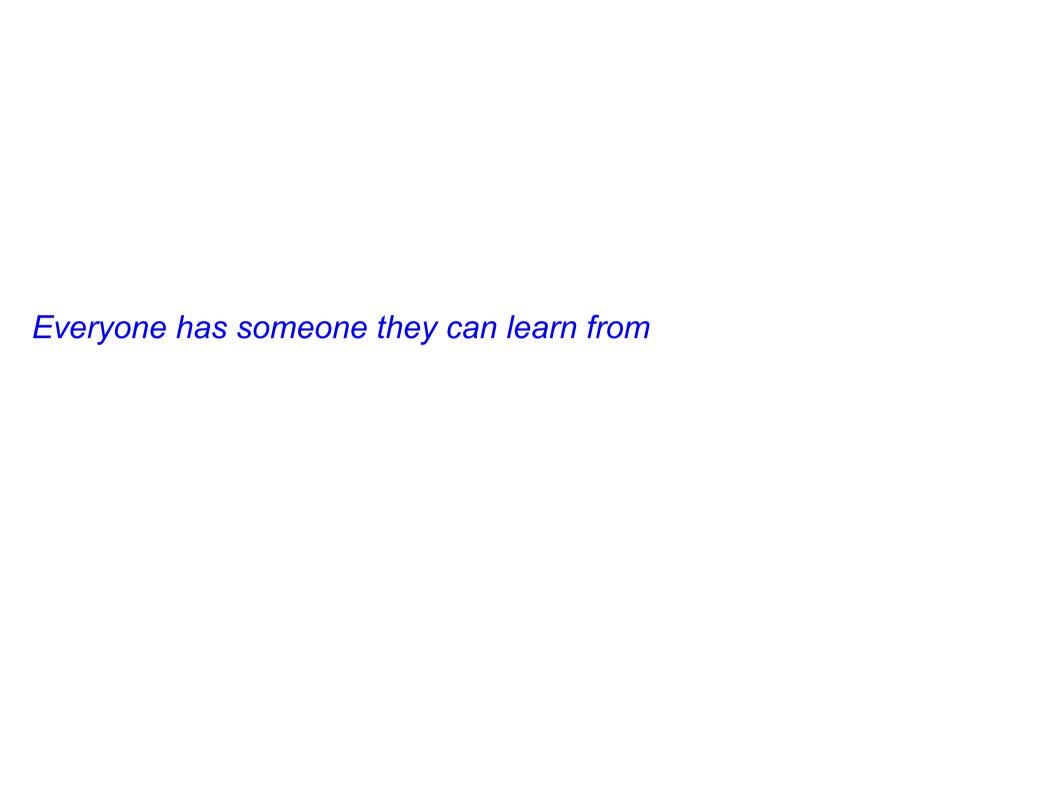
can be read as "there is a P that is also a Q" or "some P's are Q's."

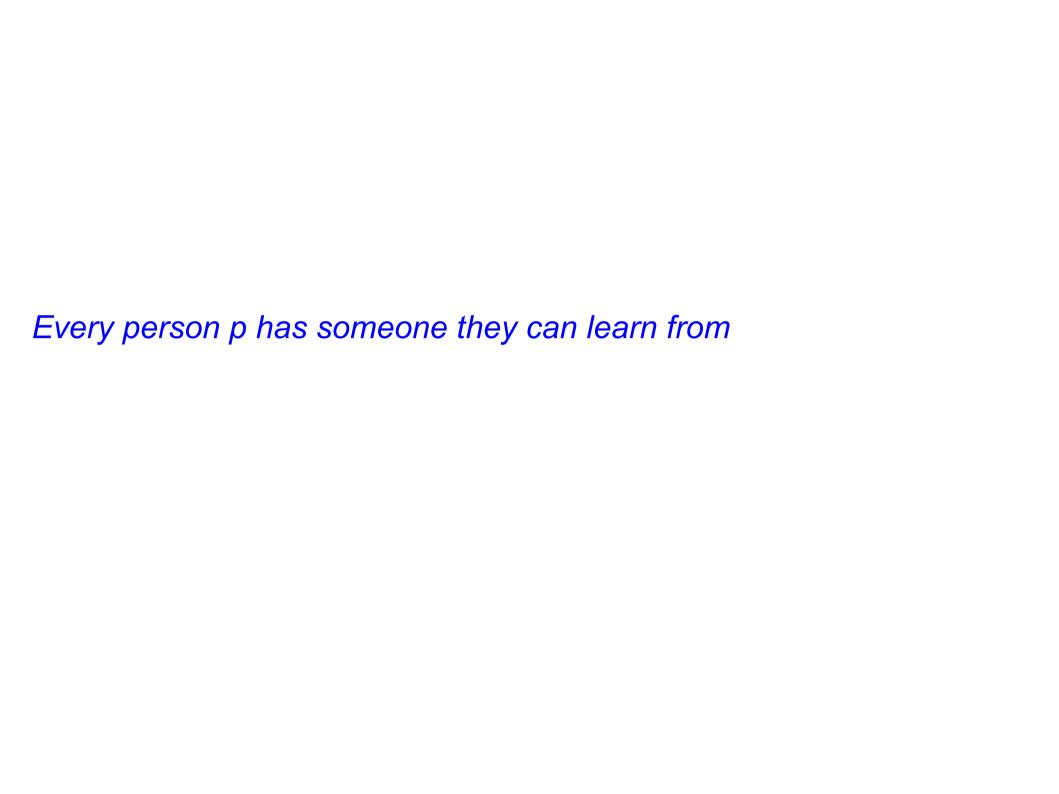
Remember: If you see ∃ paired with → or ∀ paired with ∧, the statement is probably incorrect!

Given the predicates

- $\cdot Person(p)$, which states that p is a person, and
- · 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."





Every person p has someone they can learn from

"All As are Bs." $\forall x. (A(x) \rightarrow B(x))$

```
\forall p. (Person(p) \rightarrow p \text{ has someone they can learn from})
```

```
\forall p. (Person(p) \rightarrow there is a person q that p can learn from)
```

```
\forall p. (Person(p) \rightarrow there is a person q that p can learn from)
```

"Some As are Bs."

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

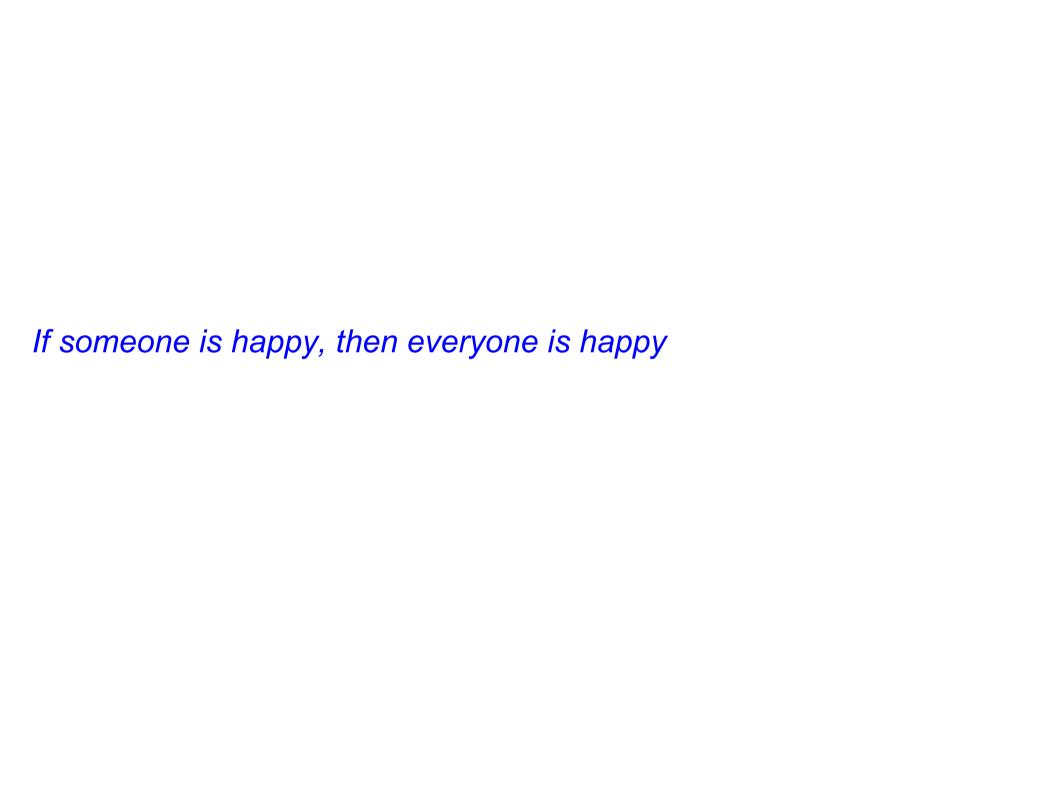
```
\forall p. (Person(p) \rightarrow \exists q. (Person(q) \land p can learn from q)
```

```
\forall p. (Person(p) \rightarrow \exists q. (Person(q) \land CanLearnFrom(p, q))
```

Consider this statement:

"If someone is happy, then everyone is happy."

What is the *contrapositive* of this statement?



someone is happy → everyone is happy

someone is happy \rightarrow ($\forall x. Happy(x)$)

 $\neg(\forall x. \ Happy(x)) \rightarrow \neg(\exists x. \ Happy(x))$

 $(\exists x. \neg Happy(x)) \rightarrow \neg (\exists x. Happy(x))$

"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?

 $(\exists x. \ Happy(x)) \land \neg(\forall x. \ Happy(x))$

 $(\exists x. \ Happy(x)) \land (\exists x. \ \neg Happy(x))$

 $(\exists x. \ Happy(x)) \land (\exists x. \ \neg Happy(x))$

"Someone is happy and someone is not happy."