Mathematical Logic

Part Three

Outline for Today

- **Recap from Last Time**
- **More First-Order Translations**
- **First-Order Negations**

Recap from Last Time

The Universal Quantifier

- A statement of the form $\forall x.\ \psi$ asserts that for ***every*** choice of x, the statement ψ is true.
- Examples:

$$\forall v.\ (Puppy(v) \to Cute(v))$$

$$\forall n.\ (n \in \mathbb{N} \to (Even(n) \leftrightarrow \neg Odd(n)))$$

$$Tallest(SK) \to \forall x.\ (SK \neq x \to ShorterThan(x, SK))$$

- Note the use of the $\to$ connective.

The Existential Quantifier

- A statement of the form $\exists x.\ \psi$ asserts that for ***some*** choice of x, the statement ψ is true.
- Examples:

  $\exists x.\ (Even(x) \land Prime(x))$

  $\exists x.\ (TallerThan(x, me) \land LighterThan(x, me))$

  $(\exists x.\ Appreciates(x, me)) \to Happy(me)$

- Note the use of the $\land$ connective.

Useful Intuition:

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

$$\forall x. \, (P(x) \to Q(x))$$

If x is a counterexample, it *must* have property P but not have property Q.

Useful Intuition:

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

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

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

Good Pairings

- The $\forall$ quantifier *usually* is paired with $\rightarrow$.
- The $\exists$ quantifier *usually* is paired with $\land$.
- In the case of $\forall$, the $\rightarrow$ connective prevents the statement from being *false* when speaking about some object you don't care about.
- In the case of $\exists$, the $\land$ connective prevents the statement from being *true* when speaking about some object you don't care about.

New Stuff!

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 “everybody loves someone else.”

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

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 someone that everyone else loves.”

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

Combining Quantifiers

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

$$\forall p.\ (Person(p) \to \exists q.\ (Person(q) \land p \neq q \land Loves(p, q)))$$

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.”

$$\exists p.\ (Person(p) \land \forall q.\ (Person(q) \land p \neq q \to Loves(q, p)))$$

There is some person

who everyone

who isn't them

loves.

For Comparison

$$\forall p.\ (Person(p) \to \exists q.\ (Person(q) \land p \neq q \land Loves(p, q)))$$

For every person,

there is some person

who isn't them

that they love.

$$\exists p.\ (Person(p) \land \forall q.\ (Person(q) \land p \neq q \to Loves(q, p)))$$

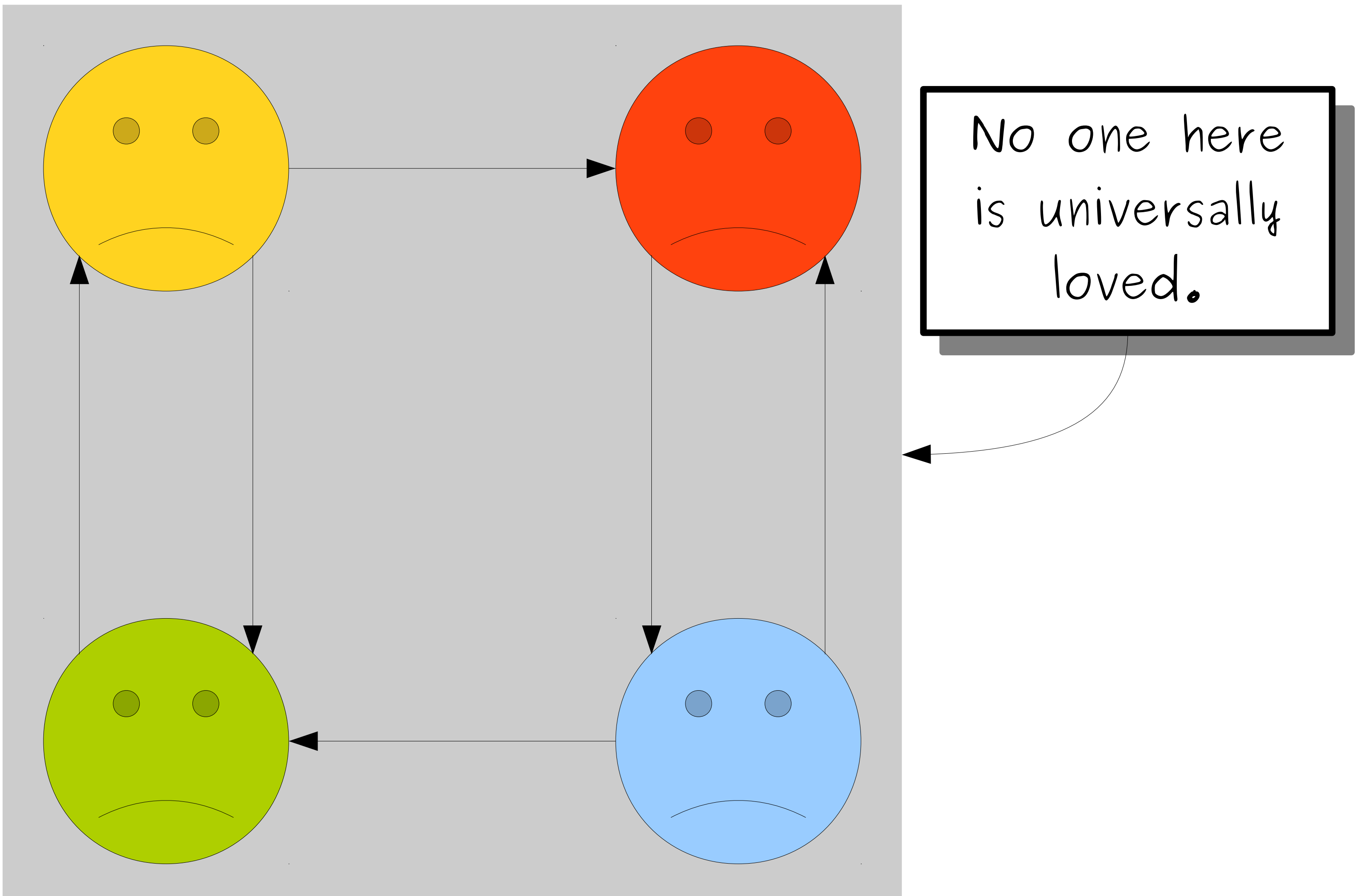
There is some person

who everyone

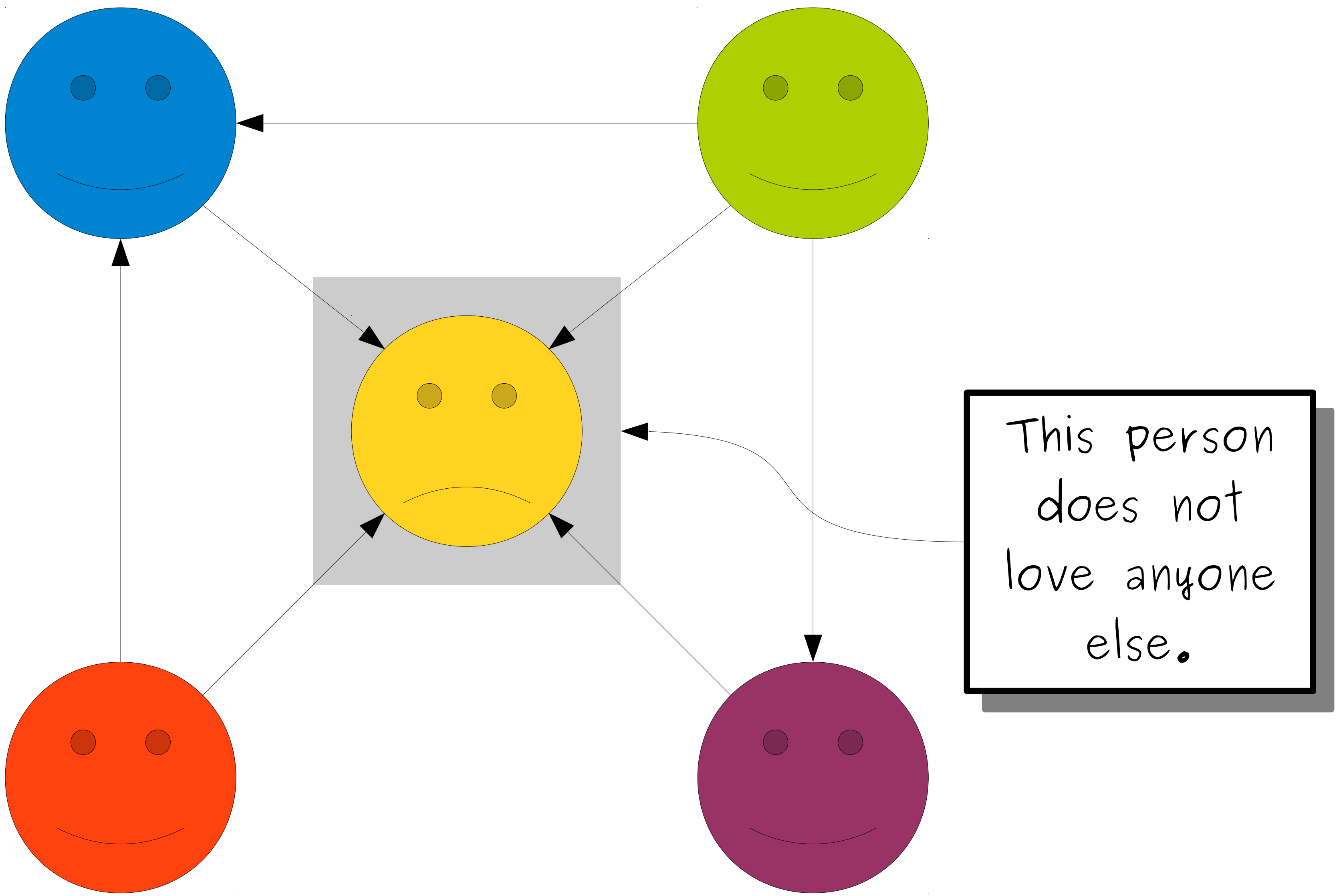
who isn't them

loves.

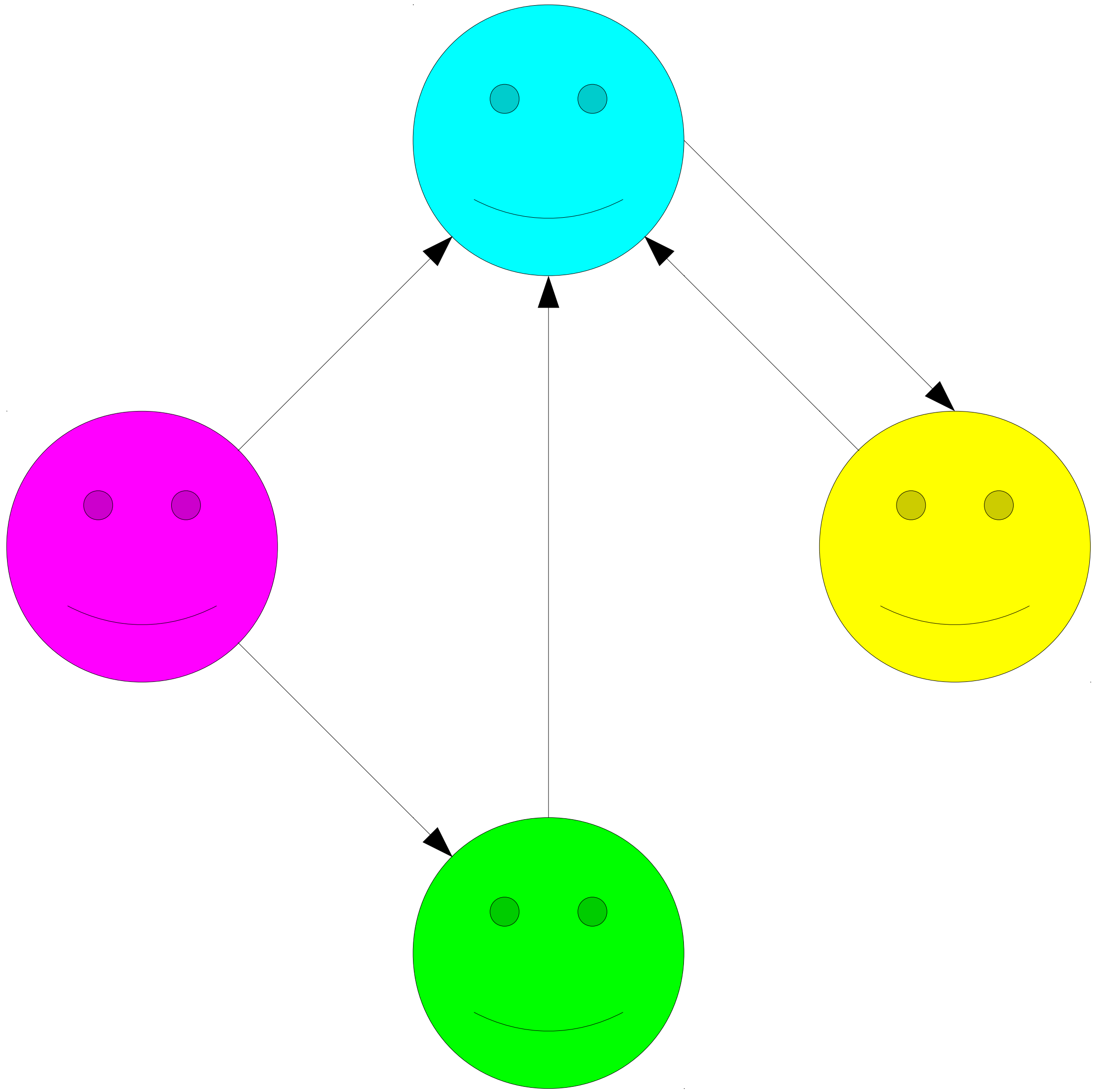
Everyone Loves Someone Else



There is Someone Everyone Else Loves



Everyone Loves Someone Else ***and*** There is Someone Everyone Else Loves



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

For every person,

there is some person

who isn't them

that they love.

$$\land$$

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

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 x, there's some 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!

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.”

First-order logic doesn't have set operators or symbols “built in.” If we only have the predicates given above, how might we describe this?

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

Using the predicates

- *Tournament*(*T*), which states that *T* is a tournament;
- $p \in T$, which states that *p* is a player in tournament *T*; and
- *Beat*(p_1, p_2), which states that p_1 beat p_2,

write a sentence in first-order logic that means “every tournament has a tournament winner.”

$$
\begin{aligned}
&\forall T.\ (Tournament(T) \rightarrow \\
&\quad \exists w.\ (w \in T \land \\
&\quad\quad \forall p.\ (p \in T \land p \neq w \rightarrow \\
&\quad\quad\quad Beat(w, p) \lor \\
&\quad\quad\quad \exists q.\ (q \in T \land \\
&\quad\quad\quad\quad Beat(w, q) \land Beat(q, p) \\
&\quad\quad\quad) \\
&\quad\quad) \\
&\quad) \\
&)
\end{aligned}
$$

Time-Out for Announcements!

PS1 Graded

- Problem Set One has been graded; feedback and grades available online via Scoryst.
- What you should do now:
  - Review all of the feedback on the problem set and make sure you understand it.
  - If you don't understand a deduction, stop by office hours, email the staff list, or ask on Piazza.
  - Incorporate that feedback when writing up your proofs for Problem Set Two.
- Regrade policy will be announced over email.

24. Albert and Bernard just become friends with Cheryl, and they want to know when her birthday is. Cheryl gives them a list of 10 possible dates.

May 15	May 16	May 19
June 17	June 18	
July 14	July 16	
August 14	August 15	August 17

Cheryl then tells Albert and Bernard separately the month and the day of her birthday respectively.

Albert: I don't know when Cheryl's birthday is, but I know that Bernard does not know too.

Bernard: At first I don't know when Cheryl's birthday is, but I know now.

Albert: Then I also know when Cheryl's birthday is.

So when is Cheryl's birthday?

Reminder: Casual CS Dinner

- WiCS is holding its first Casual CS Dinner of the quarter tonight from 6PM – 8PM on the fifth floor of Gates.
- Fantastic event; everyone is welcome and I highly recommend it.

Latin@ Coder Summit

- First ever **Latin@ Coder Summit** on May 2nd.
- Want to attend? RSVP using this link.
- Want to volunteer at the event? Click here.
- Have general questions? Contact Estefania Ortiz at eaortiz@stanford.edu.

- WiCS' annual hackathon, ***HackOverflow***, is coming up on April 25.
  - Co-hosted with DiversityBase, Stanford Robotics Club, and Stanford ACM.
- Geared toward beginners, but everyone is welcome!
- Click here to RSVP.

Your Questions

"Any thoughts on the new CS+X majors?"

They look awesome! And wonderful!
I'm so excited to see what comes
out of this program!

“What do you say to students choosing between CS and Symsys?”

They're both great programs, and it really depends on what you want to study. Symsys gives more breadth but less depth, while CS gives more depth and less breadth. You might also want to look at CS+X, minors, and coterms for more options!

“How does one go about finding an advisor in CS?”

Come talk to professors and lecturers – we don't bite!

The lecturers and intro professors know a lot about the curriculum and can give good advice about internships and CURIS. Research professors know a lot about their research areas and can help get you started in research. You can always switch advisers to get the best of all worlds!

Back to CS103!

Mechanics: Negating Statements

Negating Quantifiers

- We spent much of last Friday's lecture discussing how to negate propositional constructs.
- How do we negate quantifiers?

An Extremely Important Table

	When is this true?	When is this false?
$\forall x.\ P(x)$	For any choice of x, $P(x)$	$\exists x.\ \neg P(x)$
$\exists x.\ P(x)$	For some choice of x, $P(x)$	$\forall x.\ \neg P(x)$
$\forall x.\ \neg P(x)$	For any choice of x, $\neg P(x)$	$\exists x.\ P(x)$
$\exists x.\ \neg P(x)$	For some choice of x, $\neg P(x)$	$\forall x.\ P(x)$

Negating First-Order Statements

- Use the equivalences

$$\neg \forall x.\ \varphi \quad \equiv \quad \exists x.\ \neg \varphi$$

$$\neg \exists x.\ \varphi \quad \equiv \quad \forall x.\ \neg \varphi$$

  to negate quantifiers.
- Mechanically:
  - Push the negation across the quantifier.
  - Change the quantifier from ∀ to ∃ or vice-versa.
- Use techniques from propositional logic to negate connectives.

Taking a Negation

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

("Everyone loves someone.")

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

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

Two Useful Equivalences

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

$$\neg(p \land q) \equiv p \to \neg q$$

$$\neg(p \to q) \equiv p \land \neg q$$

- These identities are useful when negating statements involving quantifiers.
  - ∧ is used in existentially-quantified statements.
  - → is used in universally-quantified statements.
- When pushing negations across quantifiers, we *strongly recommend* using the above equivalences to keep → with ∀ and ∧ with ∃.

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:

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

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

$$\forall x.\ (Puppy(x) \to \neg Cute(x))$$

- This says "every puppy is not cute."
- Do you see why this is the negation of the original statement from both an intuitive and formal perspective?

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

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

$$\neg\exists S.\ (Set(S) \land \forall x.\ \neg(x \in S))$$
$$\forall S.\ \neg(Set(S) \land \forall x.\ \neg(x \in S))$$
$$\forall S.\ (Set(S) \to \neg\forall x.\ \neg(x \in S))$$
$$\forall S.\ (Set(S) \to \exists x.\ \neg\neg(x \in S))$$
$$\forall S.\ (Set(S) \to \exists x.\ x \in S)$$

("Every set contains at least one element")

These two statements are *not* negations of one another. Can you explain why?

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

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

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

(*“Everything is a set that contains something”*)

Remember: ∀ usually goes with →, not ∧

Next Time

- **Graphs**
  - How do we model relationships between objects?
  - How do we study the properties of those relationships?