

First-Order Logic

Part Two

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.

$\exists m. (Muggle(m) \wedge Intelligent(m))$

\exists is the **existential quantifier** and says "for some choice of m , the following is true."

“For any natural number n ,
 n is even if and only if n^2 is even”

$\forall n. (n \in \mathbb{N} \rightarrow (Even(n) \leftrightarrow Even(n^2)))$

\forall is the **universal quantifier**
and says “for any choice of n ,
the following is true.”

“Some A is a B ”

translates as

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

Useful Intuition:

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

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

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

“All A's are B's”

translates as

$\forall x. (A(x) \rightarrow B(x))$

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 .

New Stuff!

The Aristotelian Forms

“All As are Bs”

$$\forall x. (A(x) \rightarrow B(x))$$

“Some As are Bs”

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

“No As are Bs”

$$\forall x. (A(x) \rightarrow \neg B(x))$$

“Some As aren't Bs”

$$\exists x. (A(x) \wedge \neg B(x))$$

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

- $Person(p)$, 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 “every person loves someone else.”

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

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 a person that everyone else loves.”

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

Combining Quantifiers

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

For every person... $\forall p. (Person(p) \rightarrow$
... there is another person ... $\exists q. (Person(q) \wedge p \neq q \wedge$
... they love $Loves(p, q)$
)
)

Combining Quantifiers

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

There is a person... $\exists p. (Person(p) \wedge$
... that everyone else ... $\forall q. (Person(q) \wedge p \neq q \rightarrow$
... loves. $Loves(q, p))$
 $)$
 $)$

For Comparison

For every person... $\forall p. (Person(p) \rightarrow$

... there is another person ... $\exists q. (Person(q) \wedge p \neq q \wedge$
... they love $Loves(p, q)$

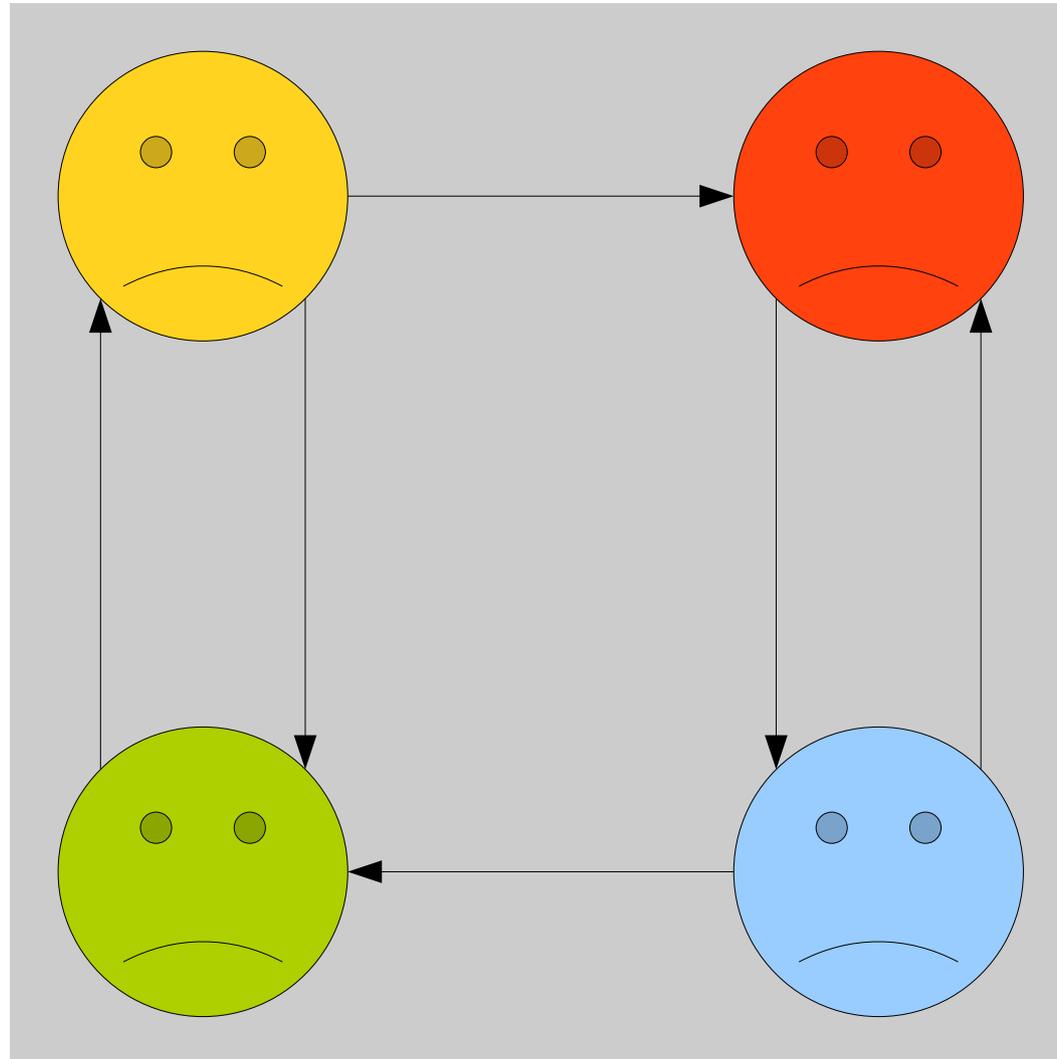
)
)

There is a person... $\exists p. (Person(p) \wedge$

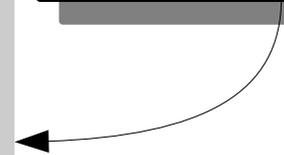
... that everyone else ... $\forall q. (Person(q) \wedge p \neq q \rightarrow$
... loves. $Loves(q, p)$

)
)

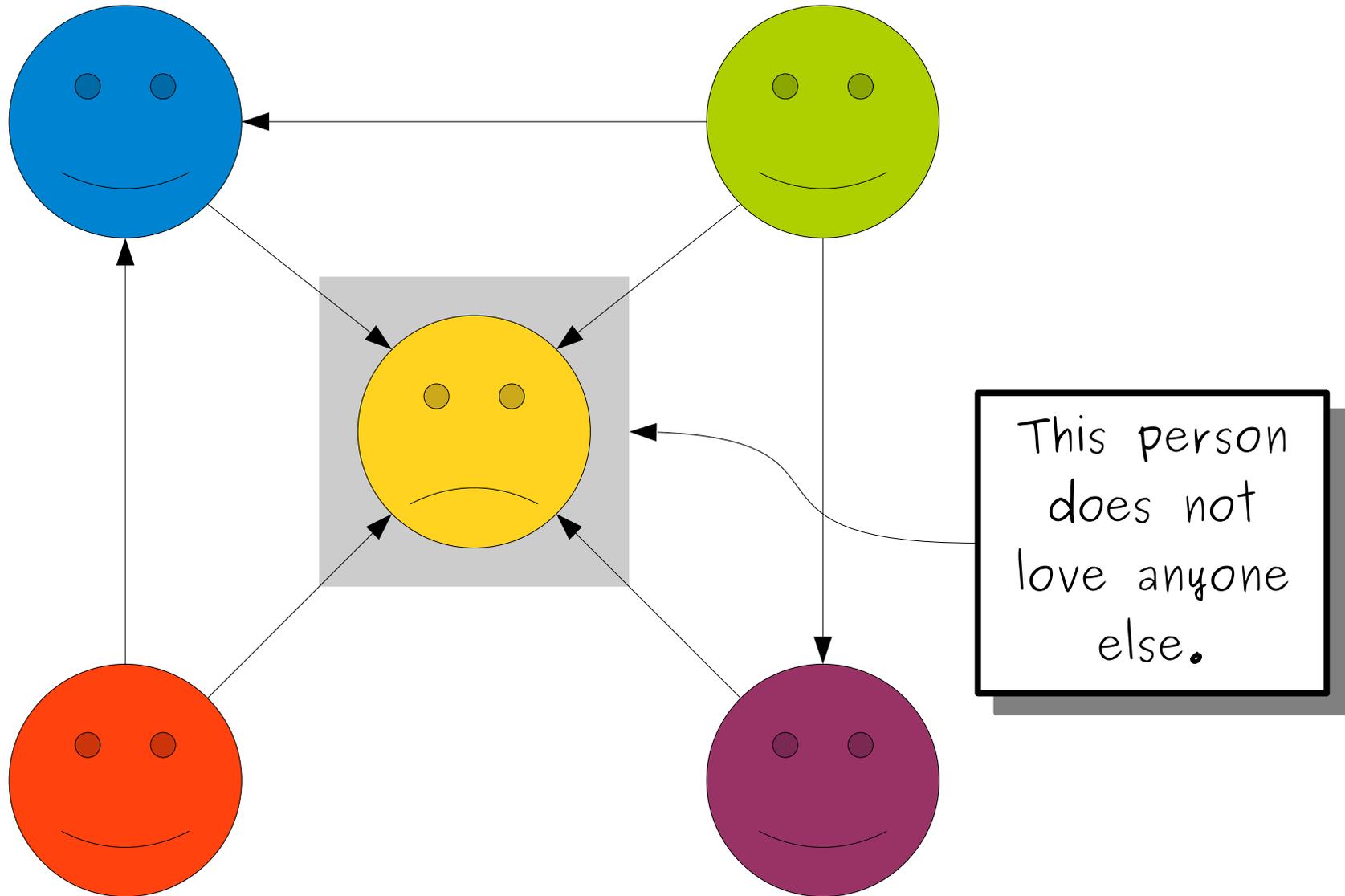
Every Person Loves Someone Else



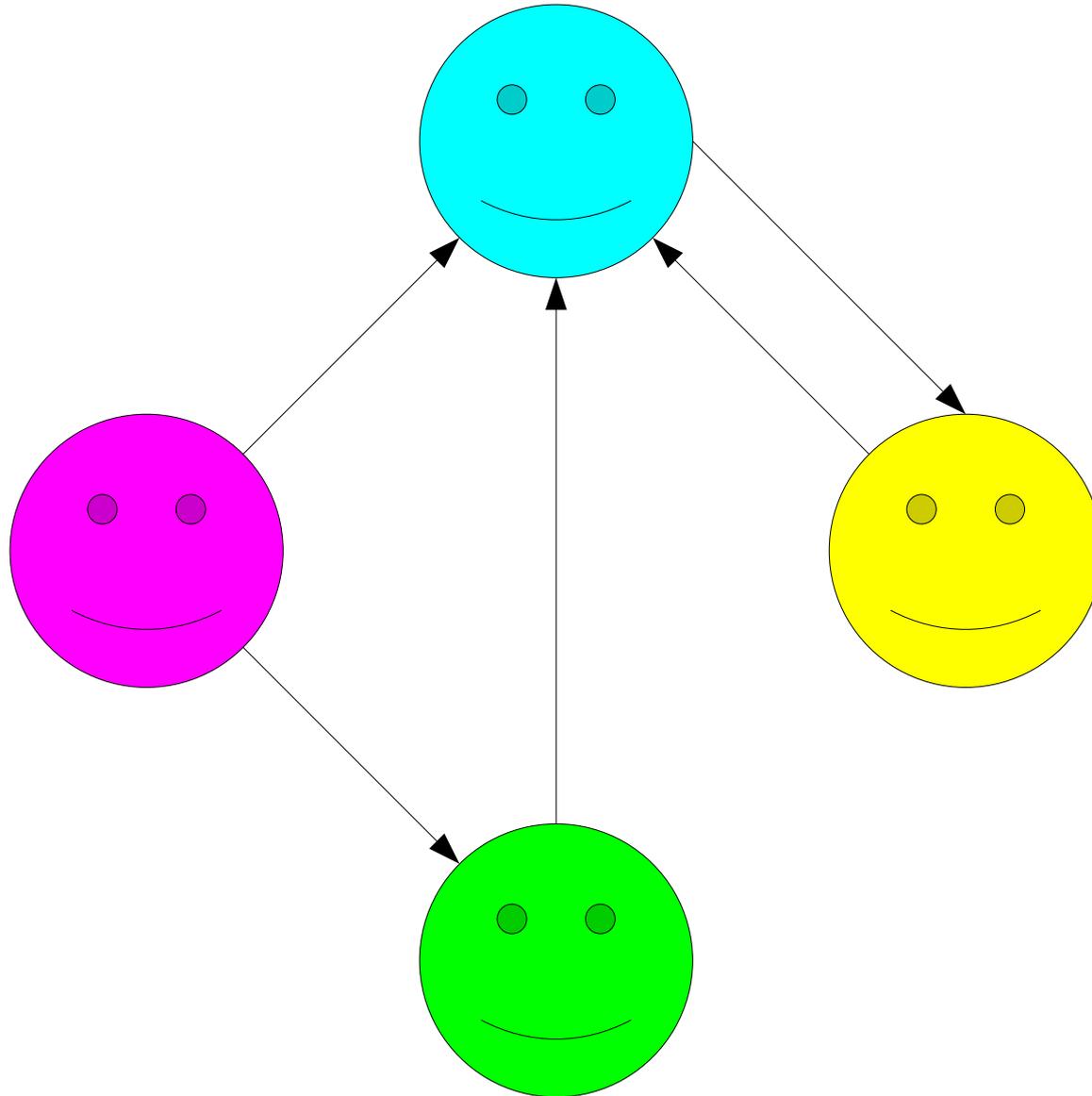
No one here is universally loved.



There is Someone Everyone Else Loves



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



For every person... $\forall p. (Person(p) \rightarrow$
 ... there is another person ... $\exists q. (Person(q) \wedge p \neq q \wedge$
 ... they love $Loves(p, q)$
)
)

and \wedge

There is a person... $\exists p. (Person(p) \wedge$
 ... that everyone else ... $\forall q. (Person(q) \wedge p \neq q \rightarrow$
 ... loves. $Loves(q, p)$
)
)

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!

Time-Out for Announcements!

COMPUTER SCIENCE

INFORMATION SESSION



October 11, 2022

3:30-4:30pm

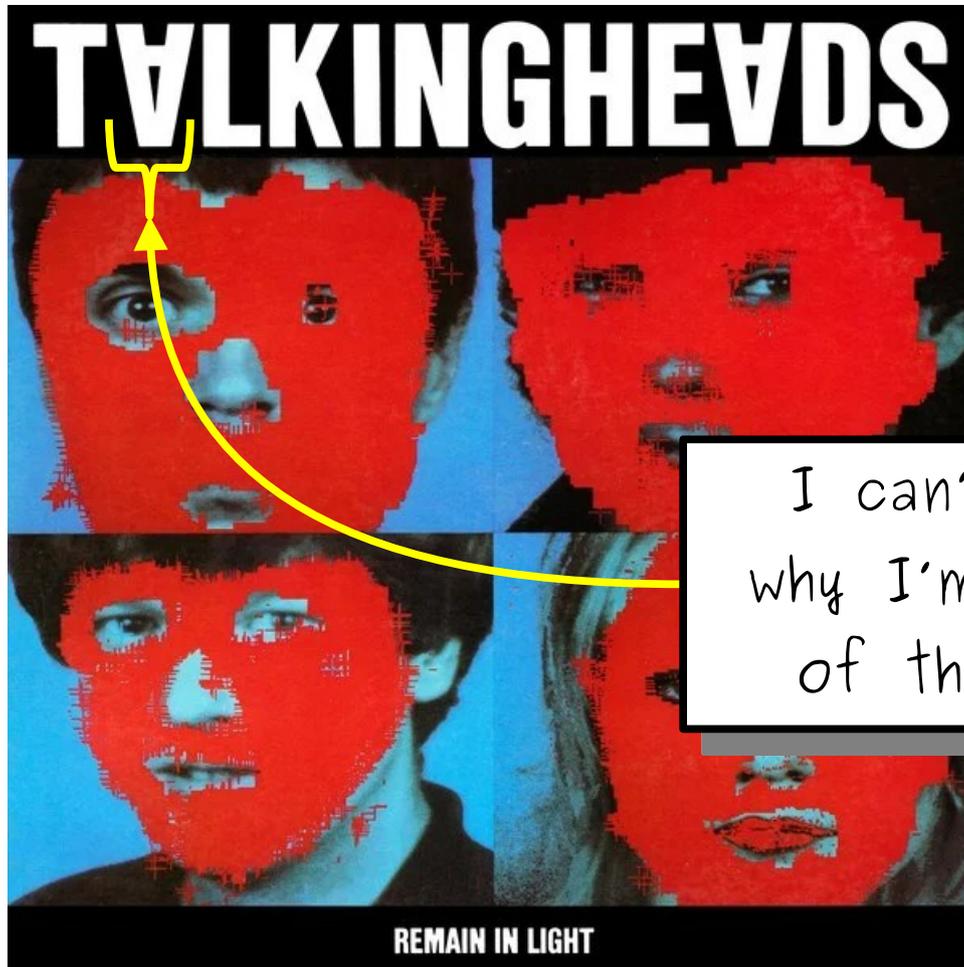
via zoom

Zoom link:

<https://tinyurl.com/3nzeveax>

Do you want to know more about declaring a major in CS or how to fill out a BSCS program sheet? Join us!

Music Recommendations



- Check out the album “Remain in Light” by Talking Heads, which blends several diverse musical styles.

I can't imagine why I'm reminded of this album...

Music Recommendations



- Check out the album “Remain in Light” by Talking Heads, which blends several diverse musical styles.
- Then, when you’re done, listen to the full-length cover album “Remain in Light” by Beninese musician Angelique Kidjo.
- Then listen to [*this interview*](#) where David Byrne of Talking Heads and Angelique Kidjo discuss their influences and experiences with the albums.

Exam Scheduling

- As a reminder, our exams are on
 - Monday, October 24th, 7PM - 10PM;
 - Monday, November 14th, 7PM - 10PM; and
 - Tuesday, December 13th, 8:30AM - 11:30AM.
- If you have a conflict with any of these dates/times, please let us know as soon as possible.
- If you have OAE accommodations and haven't forwarded your OAE letter, please do as soon as possible so we can better plan for exam season.

Problem Set Two

- Problem Set One was due today at 2:30PM.
 - Didn't submit by then? Ping us ASAP.
- Problem Set Two goes out today. It's due next Friday at 2:30PM.
 - Explore first-order logic, and expand your proofwriting repertoire.
- We have some online readings for this problem set.
 - Check out the ***Guide to Logic Translations*** for more on how to convert from English to FOL.
 - Check out the ***Guide to Negations*** for information about how to negate formulas.
 - Check out the ***First-Order Translation Checklist*** for details on how to check your work.

Your Questions

“Are certain proof methods stronger than others (ex. proof by contrapositive versus direct proofs)”

Yes! There's a branch of mathematics called "reverse mathematics" that tries to quantify this in a precise way. We know that there are some results that cannot be proven without invoking proof by contradiction or something equivalent to it, for example.

An example of this is the distinction between constructive and nonconstructive proofs.

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.”

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?

$\exists S. (Set(S) \wedge \neg \exists x. x \in S)$

$\exists S. (Set(S) \wedge \forall x. x \notin S)$

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.

$\exists S. (Set(S) \wedge \neg \exists x. x \in S)$

$\exists S. (Set(S) \wedge \forall x. x \notin S)$

Why can we switch which quantifier we're using here?

Mechanics: Negating Statements

An Extremely Important Table

	When is this true?	When is this false?
$\forall x. P(x)$	For all choices 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 all choices 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

$\neg \forall x. A$ is equivalent to $\exists x. \neg A$

$\neg \exists x. A$ is equivalent to $\forall x. \neg A$

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

$\forall x. \exists y. \text{Loves}(x, y)$
(*“Everyone loves someone.”*)

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

(*“There's someone who doesn't love anyone.”*)

Two Useful Equivalences

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

$\neg(p \wedge q)$ is equivalent to $p \rightarrow \neg q$

$\neg(p \rightarrow q)$ is equivalent to $p \wedge \neg q$

- These identities are useful when negating statements involving quantifiers.
 - \wedge 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 \wedge with \exists .

Negating Quantifiers

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

$$\exists x. (\mathit{Puppy}(x) \wedge \mathit{Cute}(x))$$

- We can obtain it as follows:

$$\neg \exists x. (\mathit{Puppy}(x) \wedge \mathit{Cute}(x))$$

$$\forall x. \neg (\mathit{Puppy}(x) \wedge \mathit{Cute}(x))$$

$$\forall x. (\mathit{Puppy}(x) \rightarrow \neg \mathit{Cute}(x))$$

- 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?

$\exists S. (Set(S) \wedge \forall x. \neg(x \in S))$
(“There is a set with no elements.”)

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

$\forall S. \neg(Set(S) \wedge \forall x. \neg(x \in S))$

$\forall S. (Set(S) \rightarrow \neg \forall x. \neg(x \in S))$

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

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

(“Every set contains at least one element.”)

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

$$\forall x \in S. P(x)$$

$$\exists x \in S. P(x)$$

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:

$$\forall x \text{ with } P(x). Q(x)$$

$$\forall y \text{ such that } P(y) \wedge Q(y). R(y).$$

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

Expressing Uniqueness

Using the predicate

- *WayToFindOut*(w), which states that w is a way to find out,

write a sentence in first-order logic that means “there is only one way to find out.”

$\exists w. (WayToFindOut(w) \wedge$
 $\forall x. (x \neq w \rightarrow \neg WayToFindOut(x))$
)

$\exists w. (WayToFindOut(w) \wedge$
 $\forall x. (WayToFindOut(x) \rightarrow x = w)$
)

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

- ***Functions***
 - How do we model transformations and pairings?
- ***First-Order Definitions***
 - Where does first-order logic come into all of this?
- ***Proofs with Definitions***
 - How does first-order logic interact with proofs?