Mathematical Logic

Part Three

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.

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

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

"For any natural number n,
n is even iff n^2 is even"

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

∀ is the ***universal quantifier*** and says "for any choice of n, the following is true."

“All *A*'s are *B*'s”

translates as

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

Useful Intuition:

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

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

If x is a counterexample, it *must* have property A but not have property B.

“Some A is a B”

translates as

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

Useful Intuition:

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

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

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

The Aristotelian Forms

"All *A*s are *B*s"

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

"Some *A*s are *B*s"

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

"No *A*s are *B*s"

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

"Some *A*s aren't *B*s"

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

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 “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 a person that everyone else loves.”

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

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.”

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

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.

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

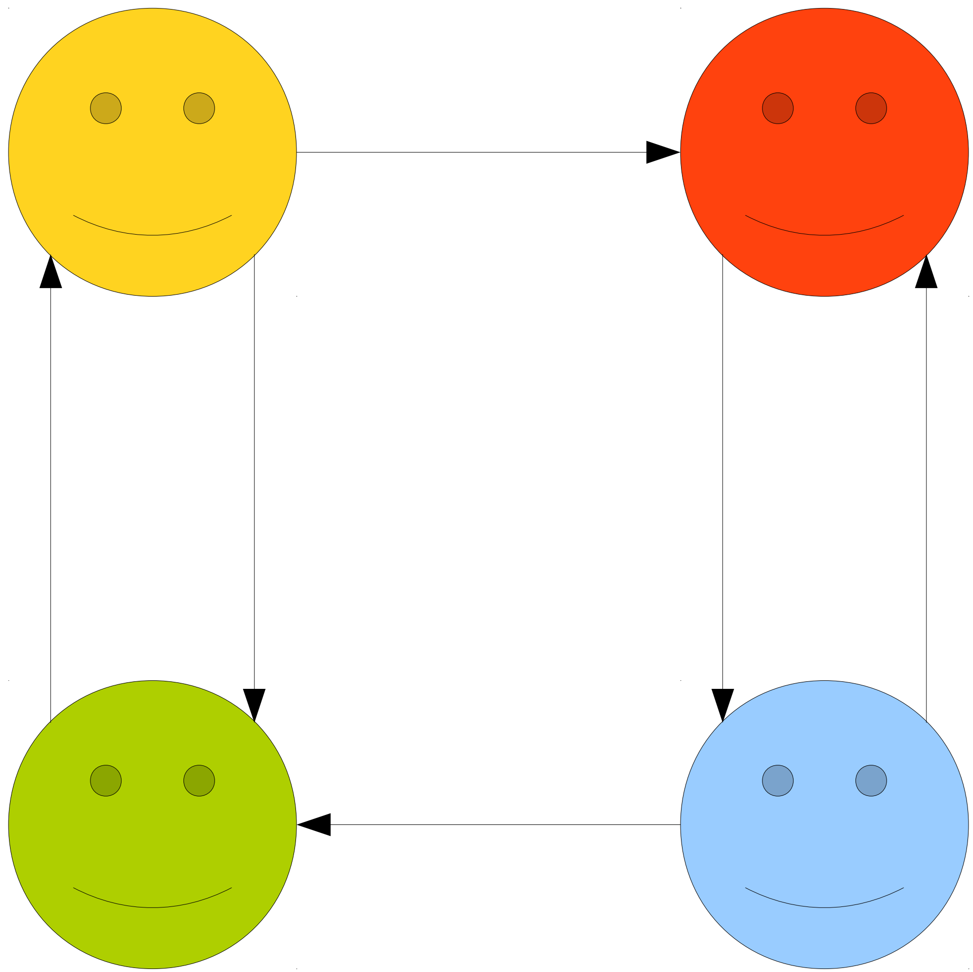
There is some person

who everyone

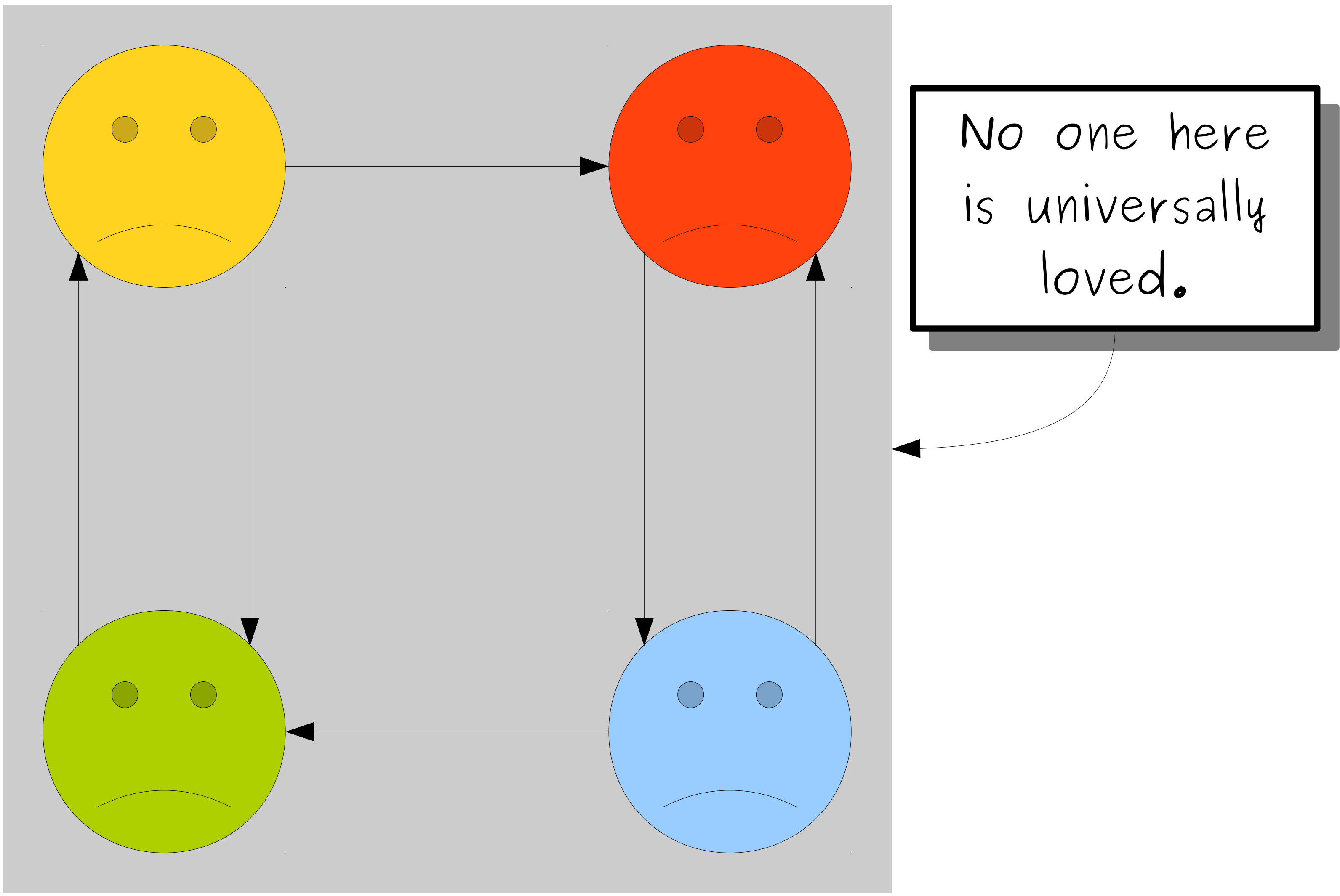
who isn't them

loves.

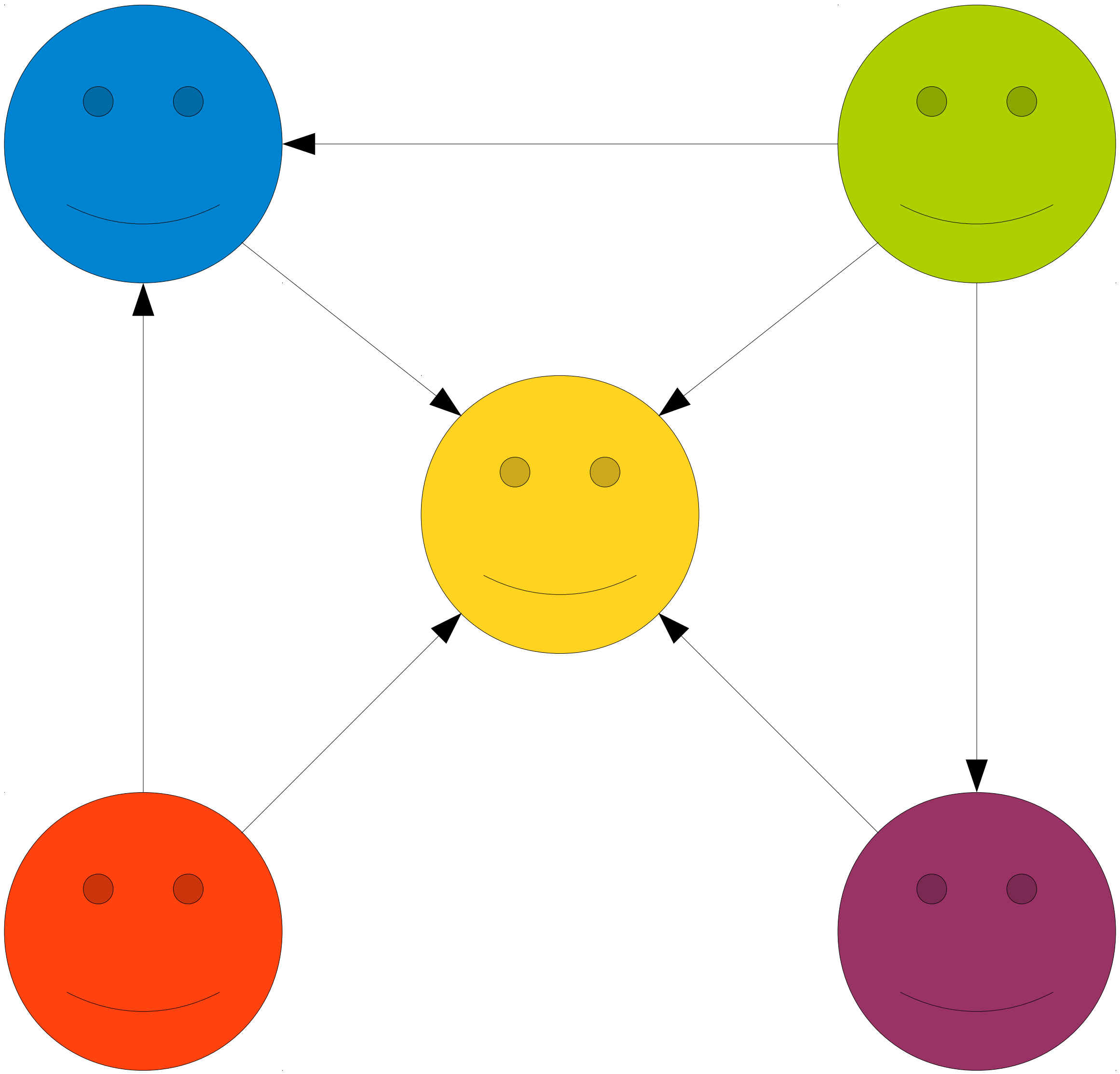
Everyone Loves Someone Else



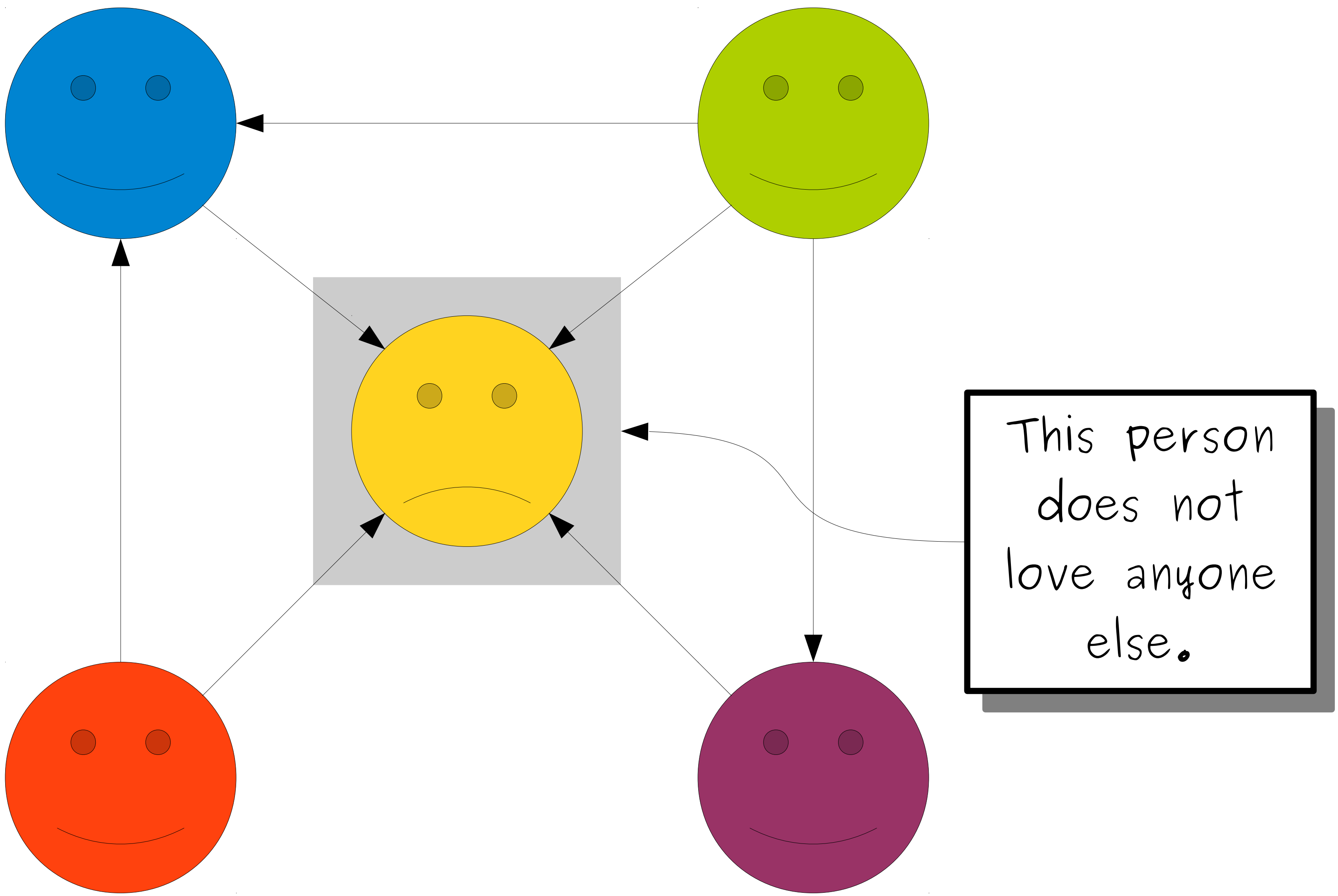
Everyone Loves Someone Else



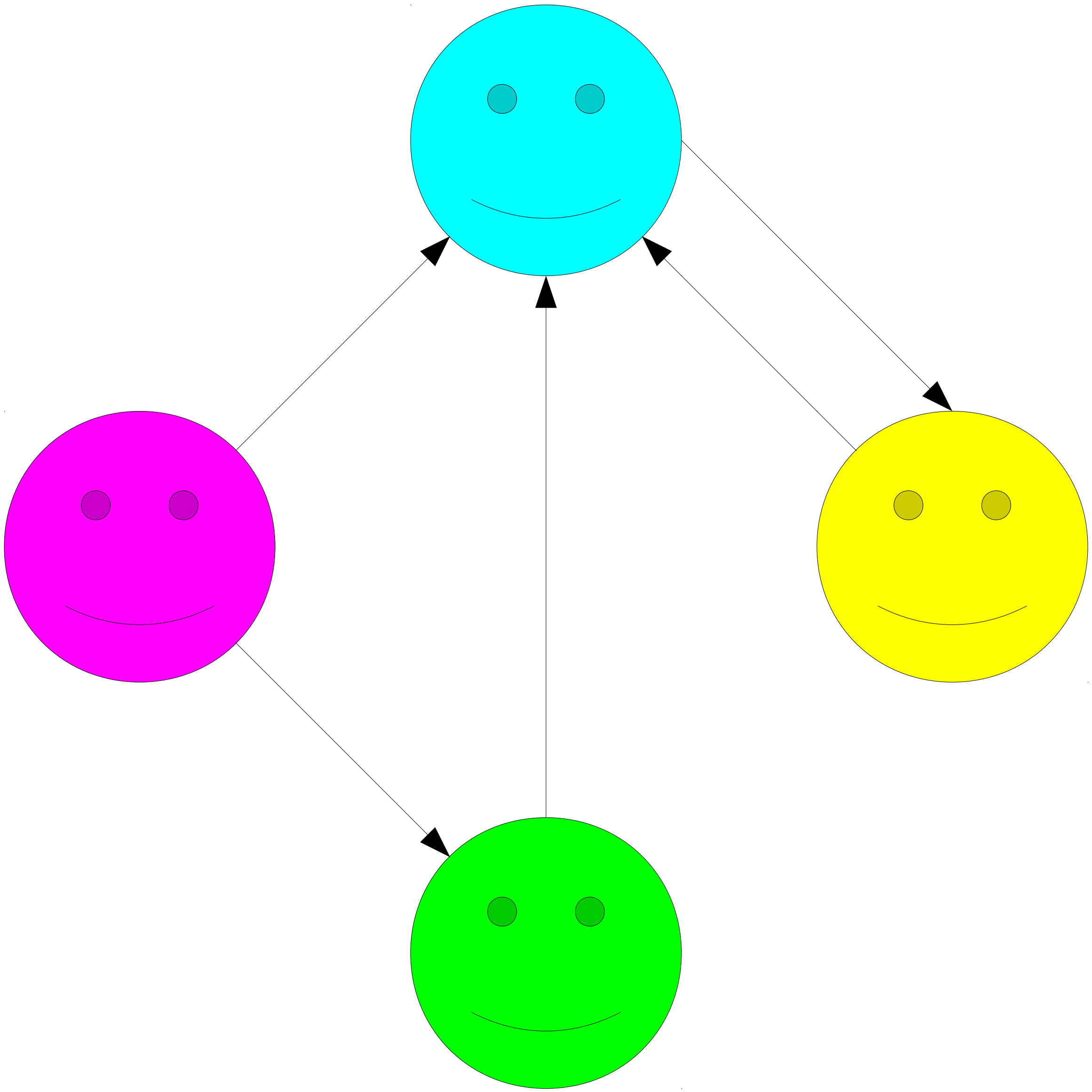
There is Someone Everyone Else Loves



There is Someone Everyone Else Loves



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



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

For every person,

there is some person

who isn't them

that they love.

$$\land$$

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

There is some person

who everyone

who isn't them

loves.

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!

Problem Set Two

- Problem Set One was due at 2:30PM today. You can submit late up until Monday at 2:30PM.
- Problem Set Two goes out now.
  - Checkpoint due Monday at 2:30PM.
  - Remaining problems due Friday.
- Play around with propositional and first-order logic and practice your proofwriting!
- As always, feel free to ask us questions in office hours or on Piazza!

Your Questions

“What do you think is the area of CS that has the highest potential to change the world?”

Designing for affordability. A huge number of people in the world have a computer in the form of a (non-smart) phone, and that's helped people get connected to one another and plan for the future better. Increasing the amount of computing power we can provide per cent, and using that to do things that matter, could radically transform the world.

“Technology has great potential to improve lives, but how do we ensure that it is addressing issues of inequity when new technologies are inherently expensive?”

At some level, technology is just something that changes cost. Think about the Bessemer process, the Model T, or the IBM PC.

I like to think about decoupling “technology” from “gadgets.” New gadgets tend to be expensive and often act as status symbols. But more broadly, many impactful technologies were huge because they were available to a mass audience.

In software, you can enable the computer to do something more cheaply than the existing alternative (e.g. Facebook). You can invent better algorithms (e.g. the fast Fourier transform or the simplex method) for solving important problems in a way that increases scalability. Or you can make better services available to a huge audience (e.g. Google Translate).

“What should I be for Halloween?”

As always, you should just be true to yourself!

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."

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?

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

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

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.

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 “two sets are equal if and only if they contain the same elements.”

$$
\begin{aligned}
&\forall S.\ (Set(S) \rightarrow \\
&\quad \forall T.\ (Set(T) \rightarrow \\
&\qquad (S = T \leftrightarrow \forall x.\ (x \in S \leftrightarrow x \in T)) \\
&\quad) \\
&)
\end{aligned}
$$

$$
\begin{aligned}
&\forall S.\ (Set(S) \rightarrow \\
&\quad \forall T.\ (Set(T) \rightarrow \\
&\quad\quad (S = T \leftrightarrow \forall x.\ (x \in S \leftrightarrow x \in T)) \\
&\quad) \\
&)
\end{aligned}
$$

You sometimes see the universal quantifier pair with the ↔ connective. This is especially common when talking about sets because two sets are equal when they have precisely the same elements.

Mechanics: Negating Statements

Negating Quantifiers

- We spent much of Monday's lecture discussing how to negate propositional constructs.
- How do we negate statements with quantifiers in them?

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

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

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

  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:

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

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

- These identities are useful when negating statements involving quantifiers.
  - $\land$ is used in existentially-quantified statements.
  - $\to$ is used in universally-quantified statements.
- When pushing negations across quantifiers, we *strongly recommend* using the above equivalences to keep $\to$ with $\forall$ and $\land$ with $\exists$.

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 “no puppy is cute.”
- Do you see why this is the negation of the original statement from both an intuitive and formal perspective?

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

(*"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: $\forall$ usually goes with $\to$, not $\land$

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

$$\forall x \in S.\ \varphi$$

$$\exists x \in S.\ \varphi$$

  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:

⚠ $\forall x$ with $P(x)$. $Q(x)$ ⚠

⚠ $\forall y$ such that $P(y) \land Q(y)$. $R(y)$. ⚠

⚠ $\exists P(x).\ Q(x)$ ⚠

Expressing Uniqueness

Using the predicate

- *Level*(l), which states that l is a level,

write a sentence in first-order logic that means "there is only one level."

A fun diversion:

http://www.onemorelevel.com/game/there_is_only_one_level

$$
\begin{aligned}
&\exists l.\ (Level(l)\ \land \\
&\quad \forall x.\ (x \neq l \rightarrow \neg Level(x)) \\
&)
\end{aligned}
$$

$$
\begin{aligned}
&\exists l.\ (Level(l)\ \wedge \\
&\quad \forall x.\ (Level(x) \rightarrow x = l) \\
&)
\end{aligned}
$$

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

- ***Binary Relations***
  - How do we model connections between objects?
- ***Equivalence Relations***
  - How do we model the idea that objects can be grouped into clusters?
- ***First-Order Definitions***
  - Where does first-order logic come into all of this?
- ***Proofs with Definitions***
  - How does first-order logic interact with proofs?