

CS103
FALL 2025



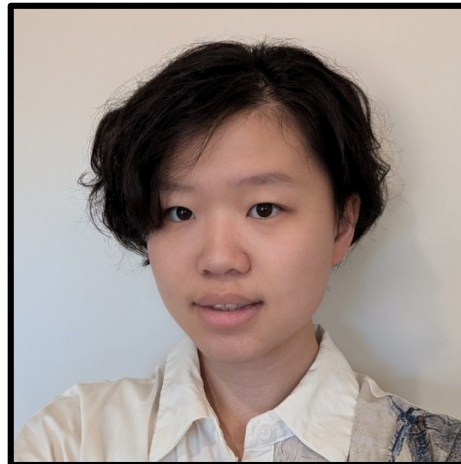
Lecture 04:

First-Order Logic

Part 1 of 2

Announcements

- Round 2 of problem set matchmaking opens today!
- We're pleased to announce Kaia Li will be our Head TA for the course!



Kaia Li
Head TA

Recap from Last Time

Recap So Far

- A **propositional variable** is a variable that is either true or false.
- The **propositional connectives** are as follows:

$\rightarrow$ $\wedge$ $\top$ $\neg$ $\vee$ $\perp$ $\leftrightarrow$

p	q	$p \rightarrow q$	$p \wedge \neg q$
F	F	T	F
F	T	T	F
T	F	F	T
T	T	T	F

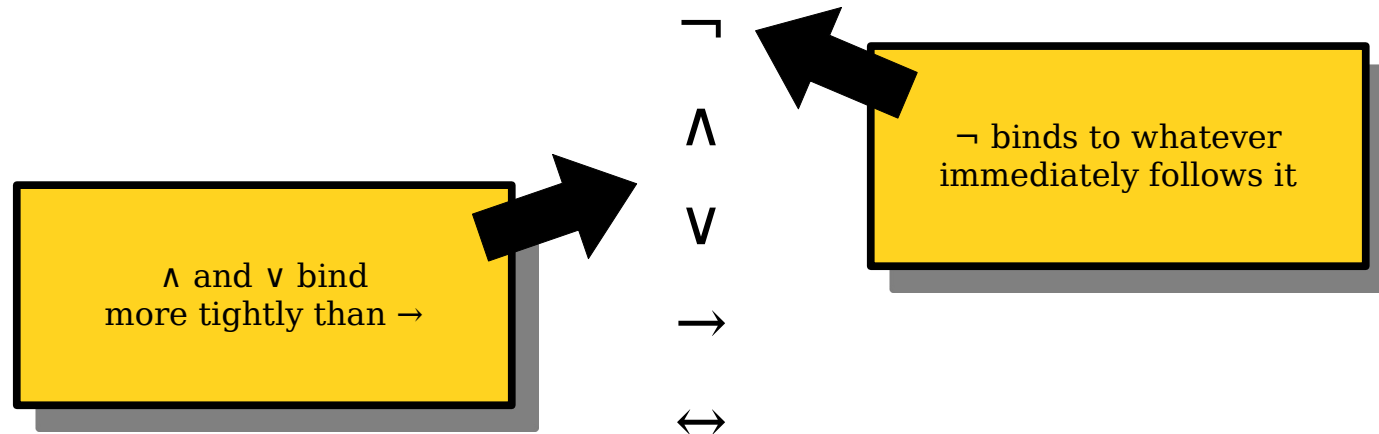
Negation of
 $p \rightarrow q$

Operator Precedence

- How do we parse this statement?

$$\neg x \rightarrow y \vee z \rightarrow x \vee y \wedge z$$

- Operator precedence for propositional logic:



- All operators are right-associative.
- We can use parentheses to disambiguate.

Why All This Matters

- Suppose we want to prove the following statement:

“If $x + y = 16$, then $x \geq 8$ or $y \geq 8$ ”

Theorem: If $x + y = 16$, then $x \geq 8$ or $y \geq 8$.

Proof: We will prove the contrapositive, namely, that if $x < 8$ and $y < 8$, then $x + y \neq 16$.

Pick x and y where $x < 8$ and $y < 8$. We want to show that $x + y \neq 16$. To see this, note that

$$\begin{aligned} x + y &< 8 + y \\ &< 8 + 8 \\ &= 16. \end{aligned}$$

This means that $x + y < 16$, so $x + y \neq 16$, which is what we needed to show. ■

New Stuff!

First-Order Logic

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 multiple objects.

Some Examples

Likes(You, Eggs) $\wedge$ Likes(You, Tomato) $\rightarrow$ Likes(You, Shakshuka)

Learns(You, History) $\vee$ ForeverRepeats(You, History)

In(MyHeart, Havana) $\wedge$ TookBackTo(Him, Me, EastAtlanta)

$Likes(You, Eggs) \wedge Likes(You, Tomato) \rightarrow Likes(You, Shakshuka)$

$Learns(You, History) \vee ForeverRepeats(You, History)$

$In(MyHeart, Havana) \wedge TookBackTo(Him, Me, EastAtlanta)$

These blue terms are called *constant symbols*. Unlike propositional variables, they refer to *objects*, not *propositions*.

Likes(You, Eggs) $\wedge$ Likes(You, Tomato) $\rightarrow$ Likes(You, Shakshuka)

Learns(You, History) $\vee$ ForeverRepeats(You, History)

In(MyHeart, Havana) $\wedge$ TookBackTo(Him, Me, EastAtlanta)

The red things that look like function calls are called *predicates*. Predicates take objects as arguments and evaluate to true or false.

Likes(You, Eggs) $\wedge$ Likes(You, Tomato) $\rightarrow$ Likes(You, Shakshuka)

Learns(You, History) $\vee$ ForeverRepeats(You, History)

In(MyHeart, Havana) $\wedge$ TookBackTo(Him, Me, EastAtlanta)

What remains are traditional propositional connectives. Because each predicate evaluates to true or false, we can connect the truth values of predicates using normal propositional connectives.

Reasoning about Objects

- To reason about objects, first-order logic uses ***predicates***.

- Examples:

Cute(Quokka)

ArgueIncessantly(Democrats, Republicans)

- Applying a predicate to arguments produces a proposition, which is either true or false.
- Typically, when you're working in FOL, you'll have a list of predicates, what they stand for, and how many arguments they take. It'll be given separately than the formulas you write.

First-Order Formulas

- Formulas in first-order logic can be constructed from predicates applied to objects:

Cute(a) → Quokka(a) ∨ Kitty(a) ∨ Puppy(a)

Succeeds(You) ↔ Practices(You)

$x < 8 \rightarrow x < 137$

The less-than sign is just another predicate. Binary predicates are sometimes written in *infix notation* this way.

Numbers are not “built in” to first-order logic. They’re constant symbols just like “You” and “a” above.

Equality

- First-order logic is equipped with a special predicate **=** that says whether two objects are equal to one another.
- Equality is a part of first-order logic, just as $\rightarrow$ and $\neg$ are.
- Examples:

TomMarvoloRiddle = LordVoldemort

MorningStar = EveningStar

- Equality can only be applied to **objects**; to state that two **propositions** are equal, use $\leftrightarrow$.

Let's see some more examples.

$\text{FavoriteMovieOf}(\text{You}) \neq \text{FavoriteMovieOf}(\text{Date}) \wedge$
 $\text{StarOf}(\text{FavoriteMovieOf}(\text{You})) = \text{StarOf}(\text{FavoriteMovieOf}(\text{Date}))$

These purple terms are *functions*. Functions take objects as input and produce objects as output.

$\text{FavoriteMovieOf}(\text{You}) \neq \text{FavoriteMovieOf}(\text{Date}) \wedge$
 $\text{StarOf}(\text{FavoriteMovieOf}(\text{You})) = \text{StarOf}(\text{FavoriteMovieOf}(\text{Date}))$

Functions

- First-order logic allows **functions** that return objects associated with other objects.
- Examples:

ColorOf(Money)

MedianOf(x, y, z)

$x + y$

- As with predicates, functions can take in any number of arguments, but always return a single value.
- Functions evaluate to **objects**, not **propositions**.

Objects and Propositions

- When working in first-order logic, be careful to keep objects (actual things) and propositions (true or false) separate.
- You cannot apply connectives to objects:

Venus $\rightarrow$ *TheSun*

- You cannot apply functions to propositions:

StarOf(IsRed(Sun) $\wedge$ IsGreen(Mars))

- Ever get confused? *Just ask!*

The Type-Checking Table

	... operate on ...	... and produce
Connectives ($\leftrightarrow$, $\wedge$, etc.) ...	propositions	a proposition
Predicates ($=$, etc.) ...	objects	a proposition
Functions ...	objects	an object

One last (and major) change

Some bear is curious.

$\exists b. (Bear(b) \wedge Curious(b))$

$\exists$ is the **existential quantifier**
and says "there is a choice of
b where the following is
true."

The Existential Quantifier

- A statement of the form

$\exists x.$ *some-formula*

is true when there exists a choice object where ***some-formula*** is true when that object is plugged in for x .

- Examples:

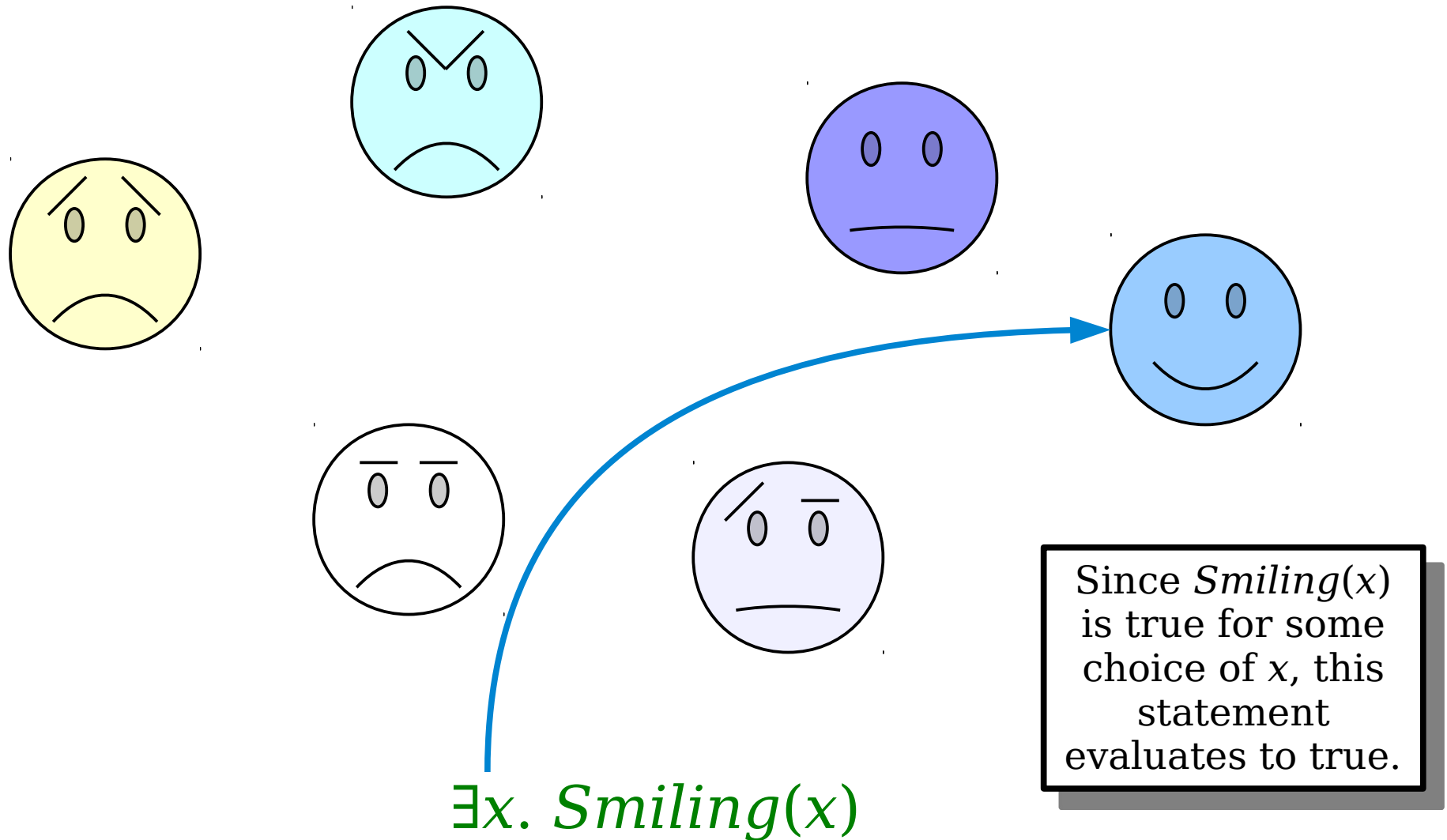
$\exists x. (Even(x) \wedge Prime(x))$

$\exists x. (TallerThan(x, me) \wedge WeighsLessThan(x, me))$

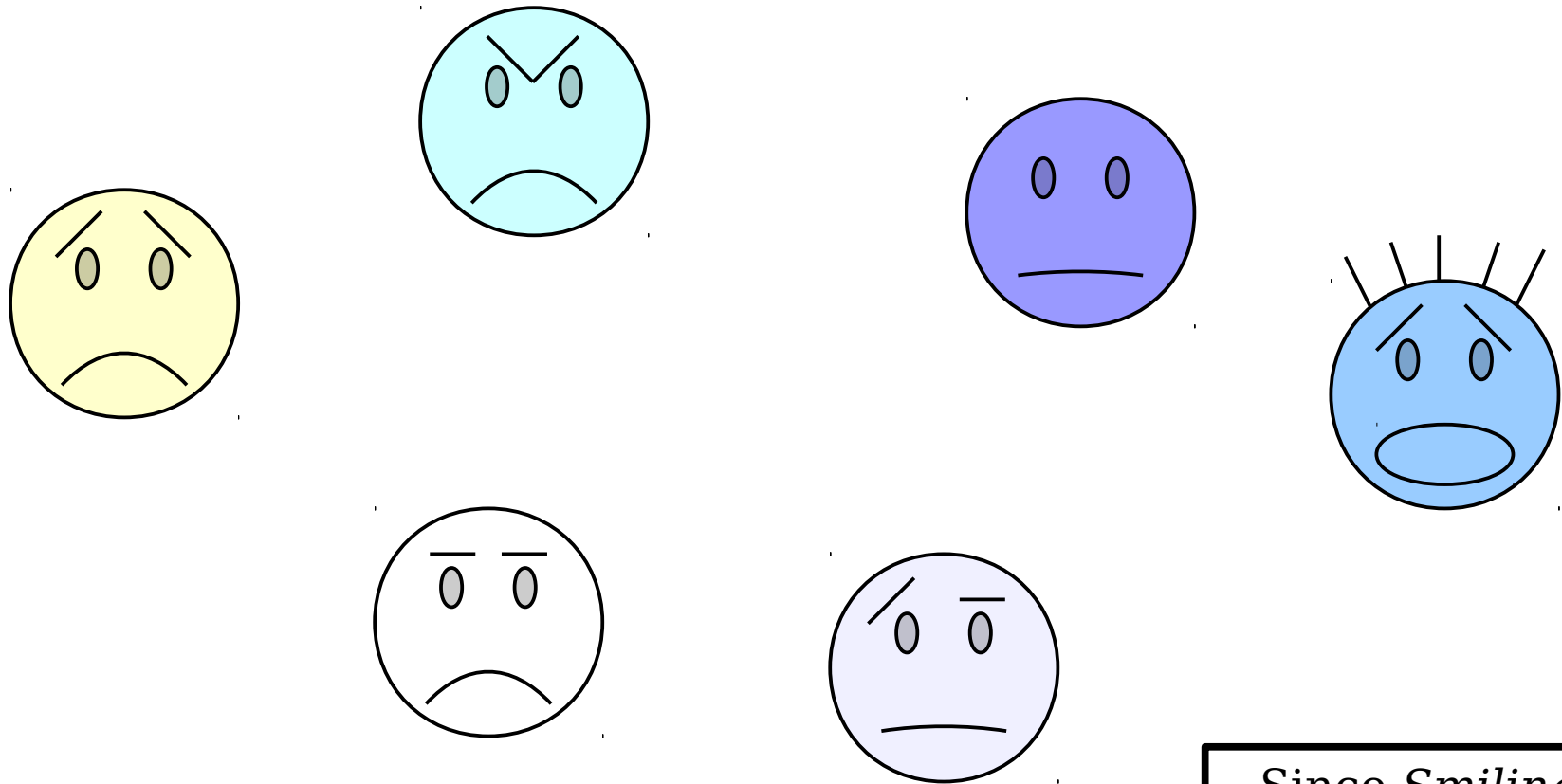
$(\exists w. Will(w)) \rightarrow (\exists x. Way(x))$

- Note the two ways of applying the $\exists$!

The Existential Quantifier



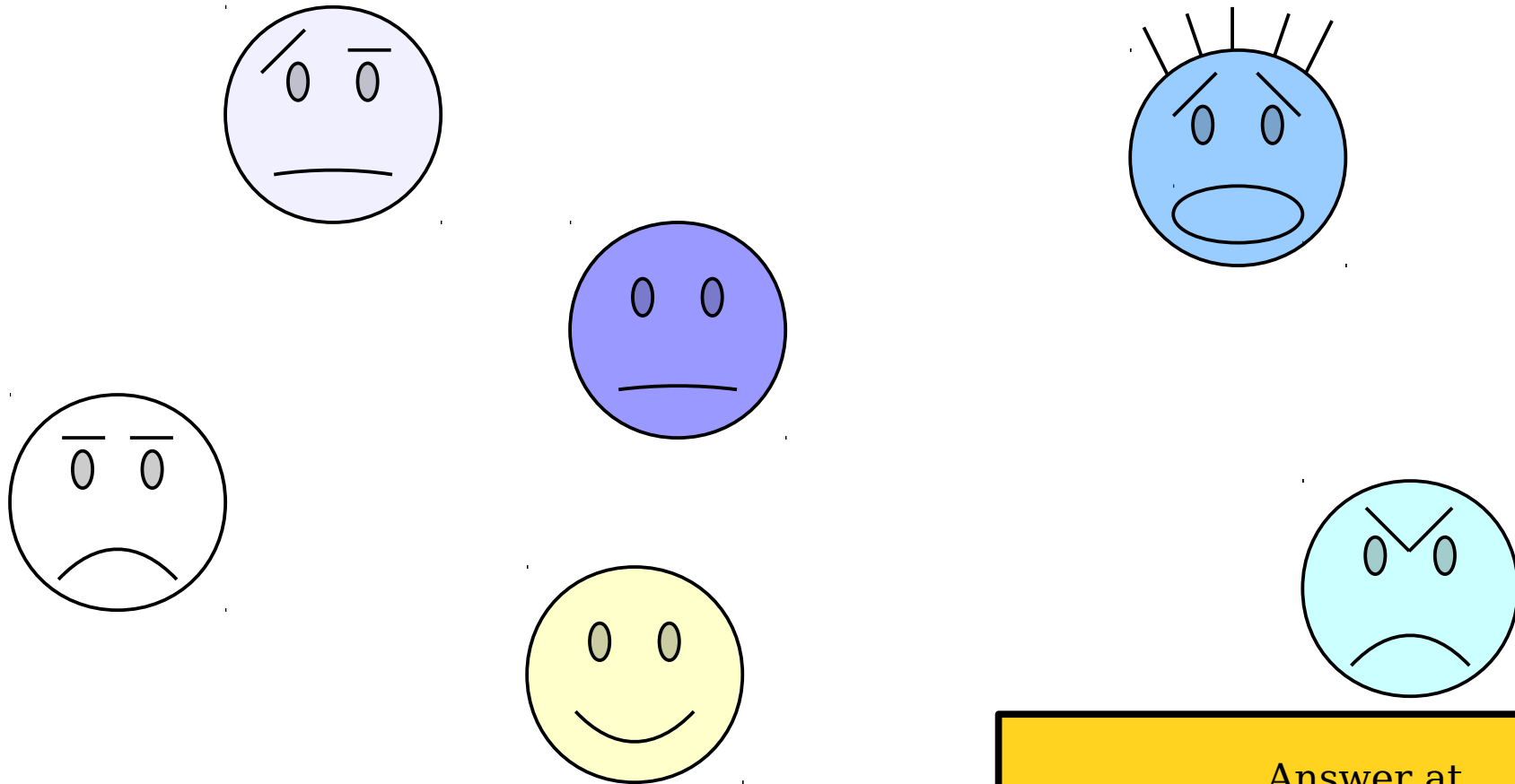
The Existential Quantifier



~~$\exists x. Smiling(x)$~~

Since $Smiling(x)$ is not true for any choice of x , this statement evaluates to false.

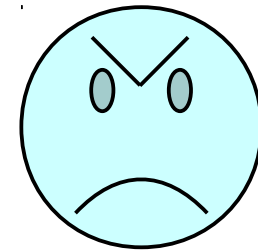
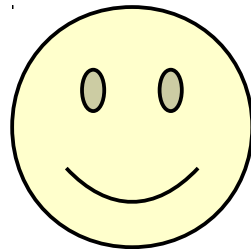
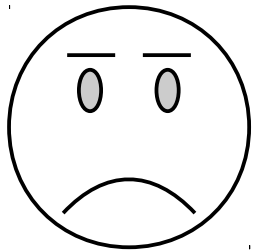
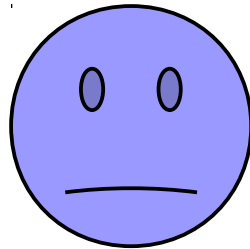
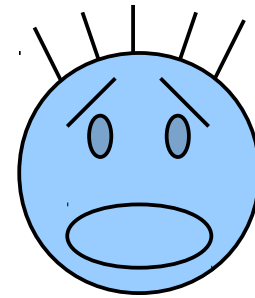
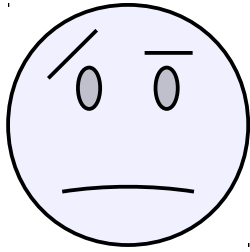
The Existential Quantifier



Answer at
<https://cs103.stanford.edu/pollev>

$$(\exists x. \textit{Smiling}(x)) \rightarrow (\exists y. \textit{WearingHat}(y))$$

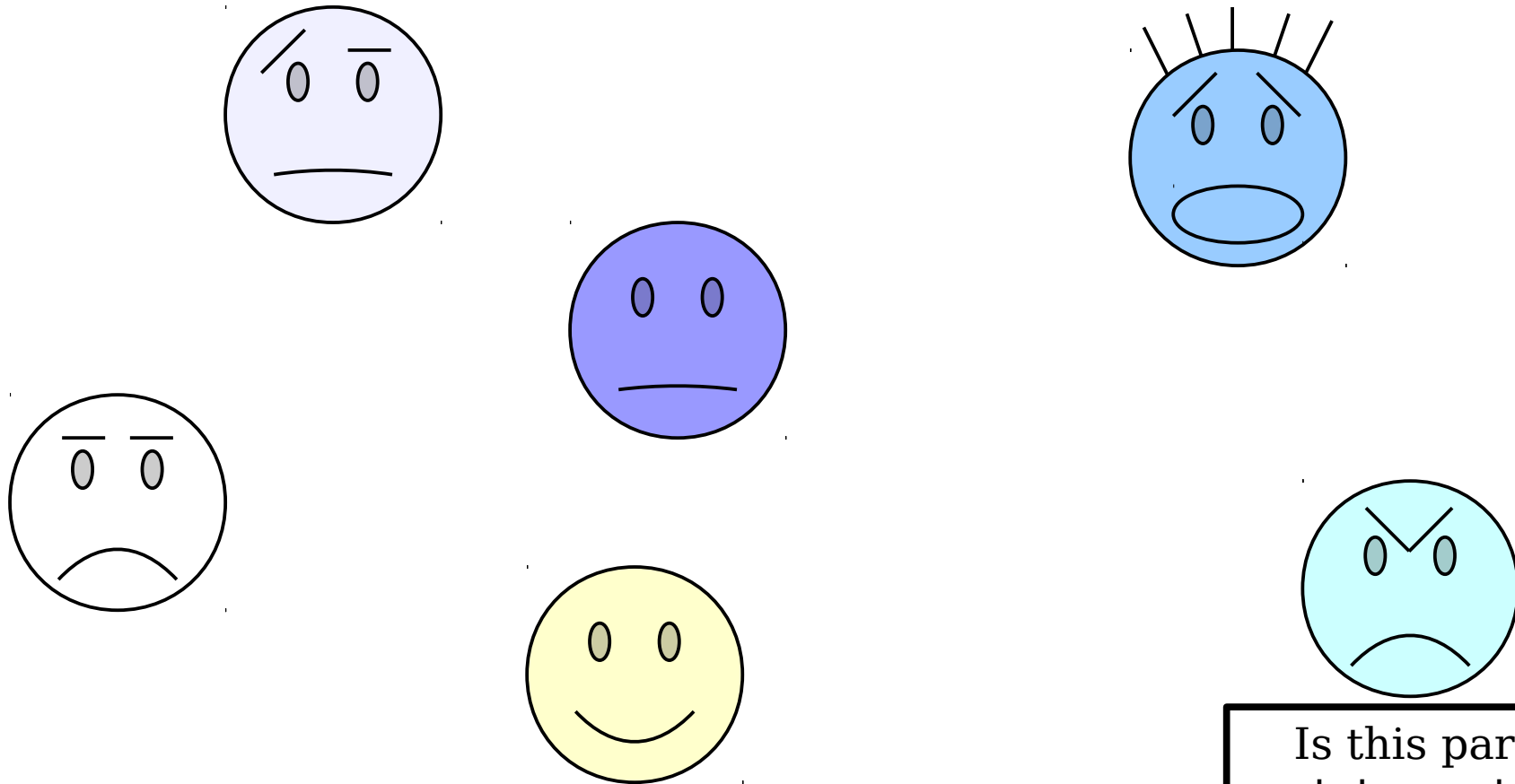
The Existential Quantifier



Is this part of the
statement true or
false?

$(\exists x. \textit{Smiling}(x)) \rightarrow (\exists y. \textit{WearingHat}(y))$

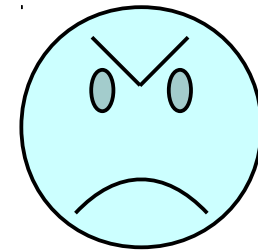
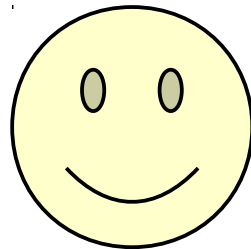
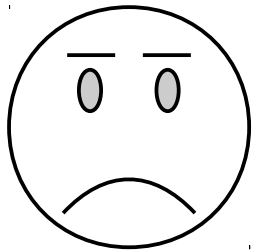
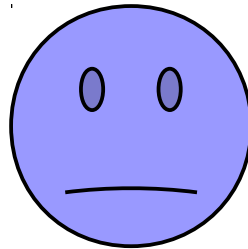
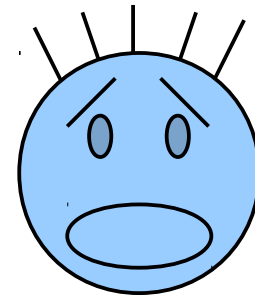
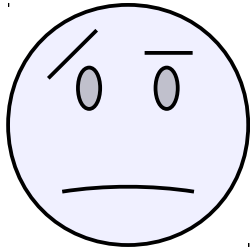
The Existential Quantifier



Is this part of the statement true or false?

$(\exists x. \textit{Smiling}(x)) \rightarrow (\exists y. \textit{WearingHat}(y))$

The Existential Quantifier



Is this overall
statement true or
false?

~~$(\exists x. \textit{Smiling}(x)) \rightarrow (\exists y. \textit{WearingHat}(y))$~~

Fun with Edge Cases

Existentially-quantified statements are false in an empty world, since nothing exists, period!

~~$\exists x. \textit{Smiling}(x)$~~

Some Technical Details

Variables and Quantifiers

- Each quantifier has two parts:
 - the variable that is introduced, and
 - the statement that's being quantified.
- The variable introduced is scoped just to the statement being quantified.

$(\exists x. \text{Loves}(\text{You}, x)) \wedge (\exists y. \text{Loves}(y, \text{You}))$

The variable x
just lives here.

The variable y
just lives here.

Variables and Quantifiers

- Each quantifier has two parts:
 - the variable that is introduced, and
 - the statement that's being quantified.
- The variable introduced is scoped just to the statement being quantified.

$$(\exists x. \text{Loves}(\text{You}, x)) \wedge (\exists x. \text{Loves}(x, \text{You}))$$

The variable x
just lives here.

A different variable,
also named x , just
lives here.

Operator Precedence (Again)

- When writing out a formula in first-order logic, quantifiers have precedence just below $\neg$.
- The statement

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

is parsed like this:

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

- This is syntactically invalid because the variable x is out of scope in the back half of the formula.
- To ensure that x is properly quantified, explicitly put parentheses around the region you want to quantify:

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

“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 all choices of n ,
the following is true.”

The Universal Quantifier

- A statement of the form

$\forall x.$ *some-formula*

is true when, for every choice of x , the statement ***some-formula*** is true when x is plugged into it.

- Examples:

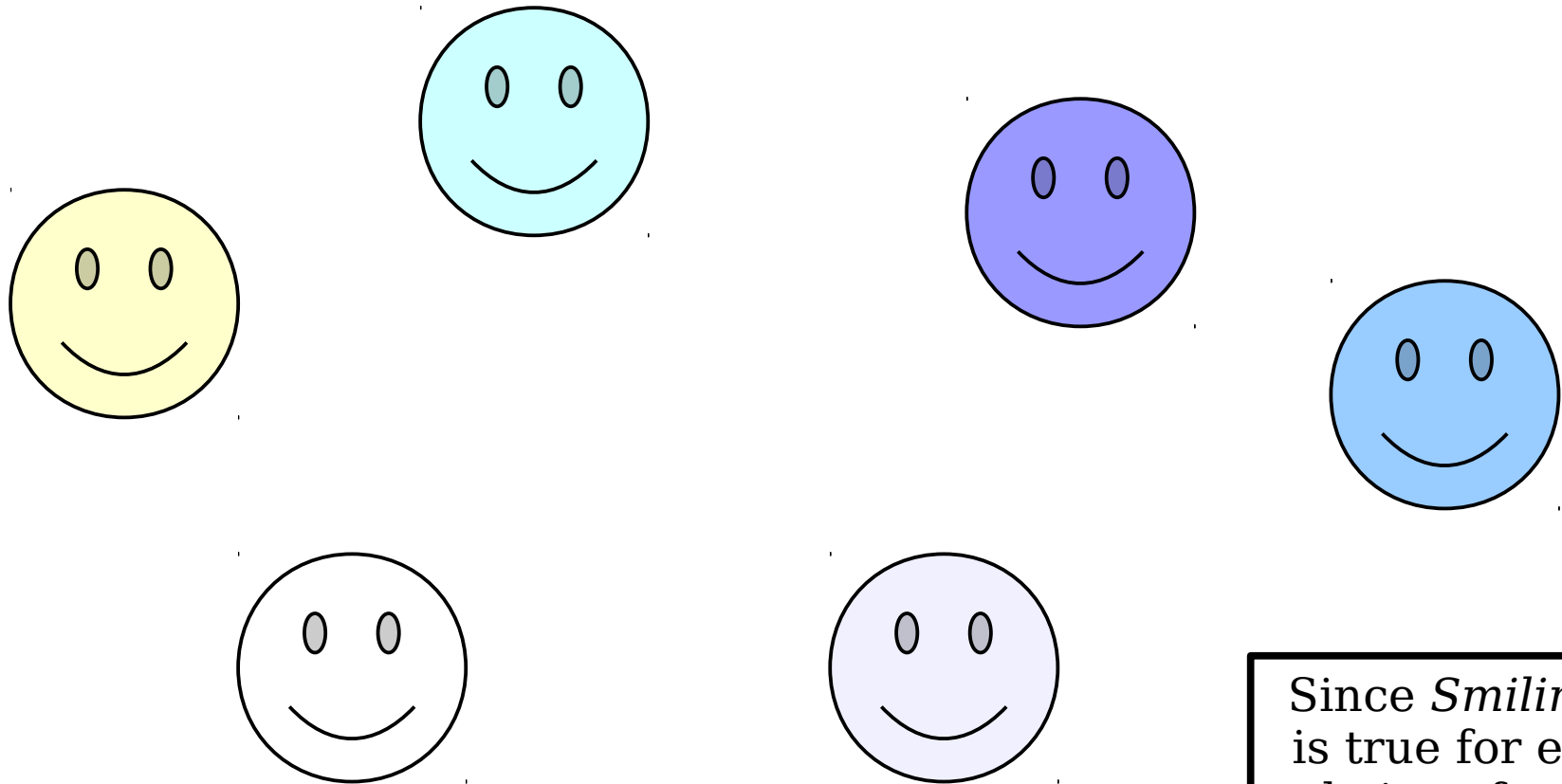
$\forall p. (Puppy(p) \rightarrow Cute(p))$

$\forall a. (EatsPlants(a) \vee EatsAnimals(a))$

$Tallest(SultanKösen) \rightarrow$

$\forall x. (SultanKösen \neq x \rightarrow ShorterThan(x, SultanKösen))$

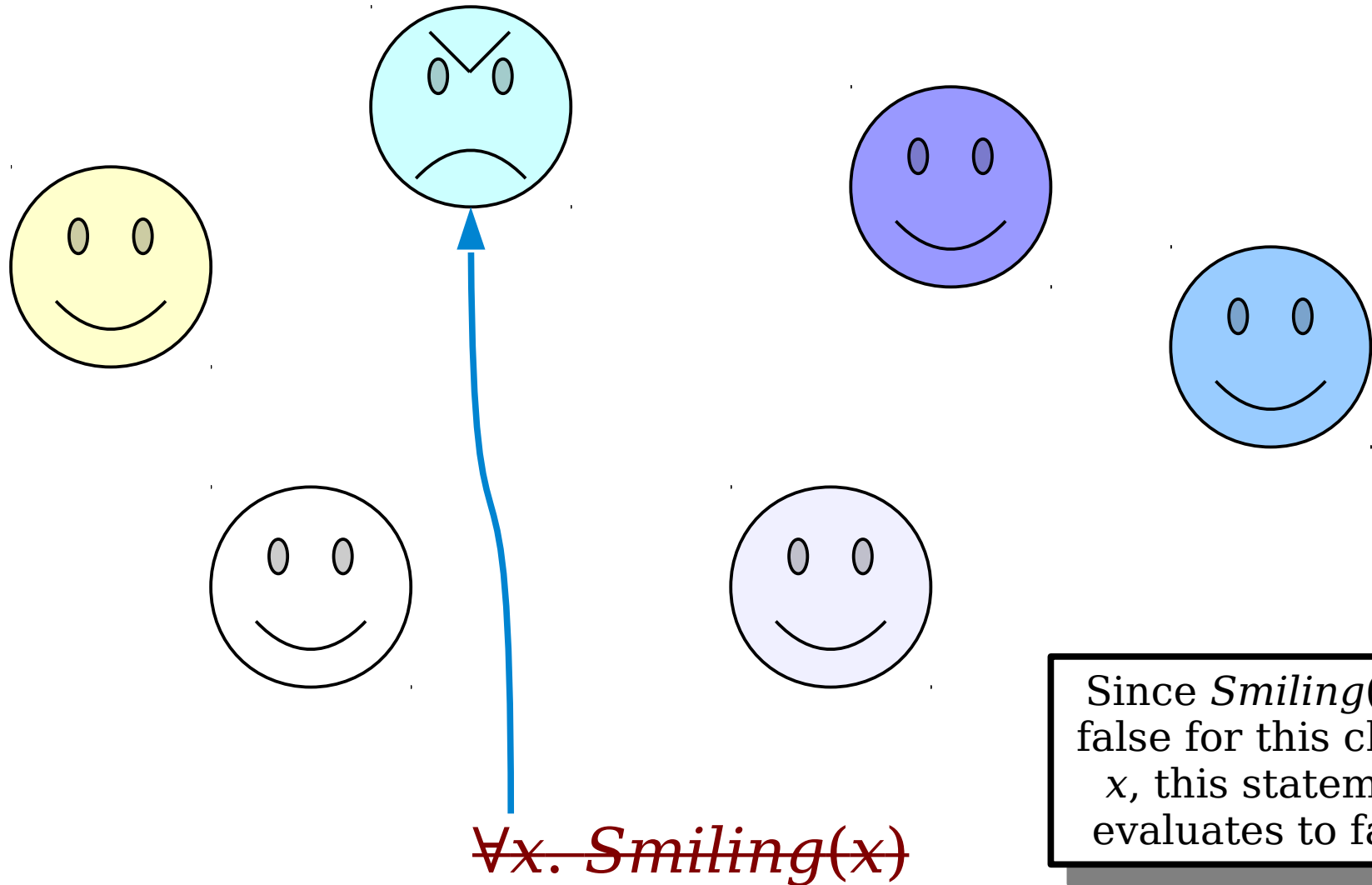
The Universal Quantifier



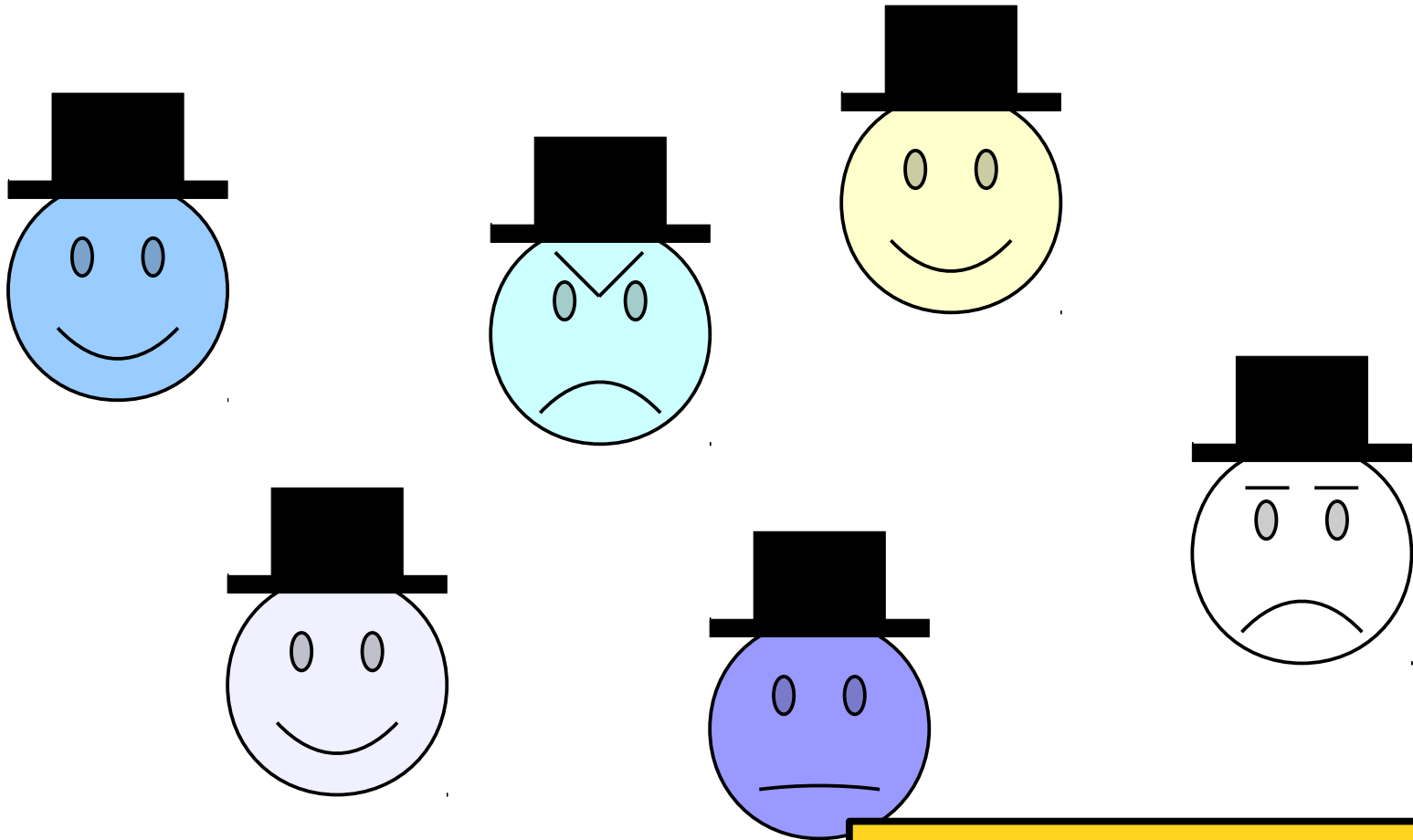
$\forall x. \textit{Smiling}(x)$

Since *Smiling*(*x*)
is true for every
choice of *x*, this
statement
evaluates to true.

The Universal Quantifier



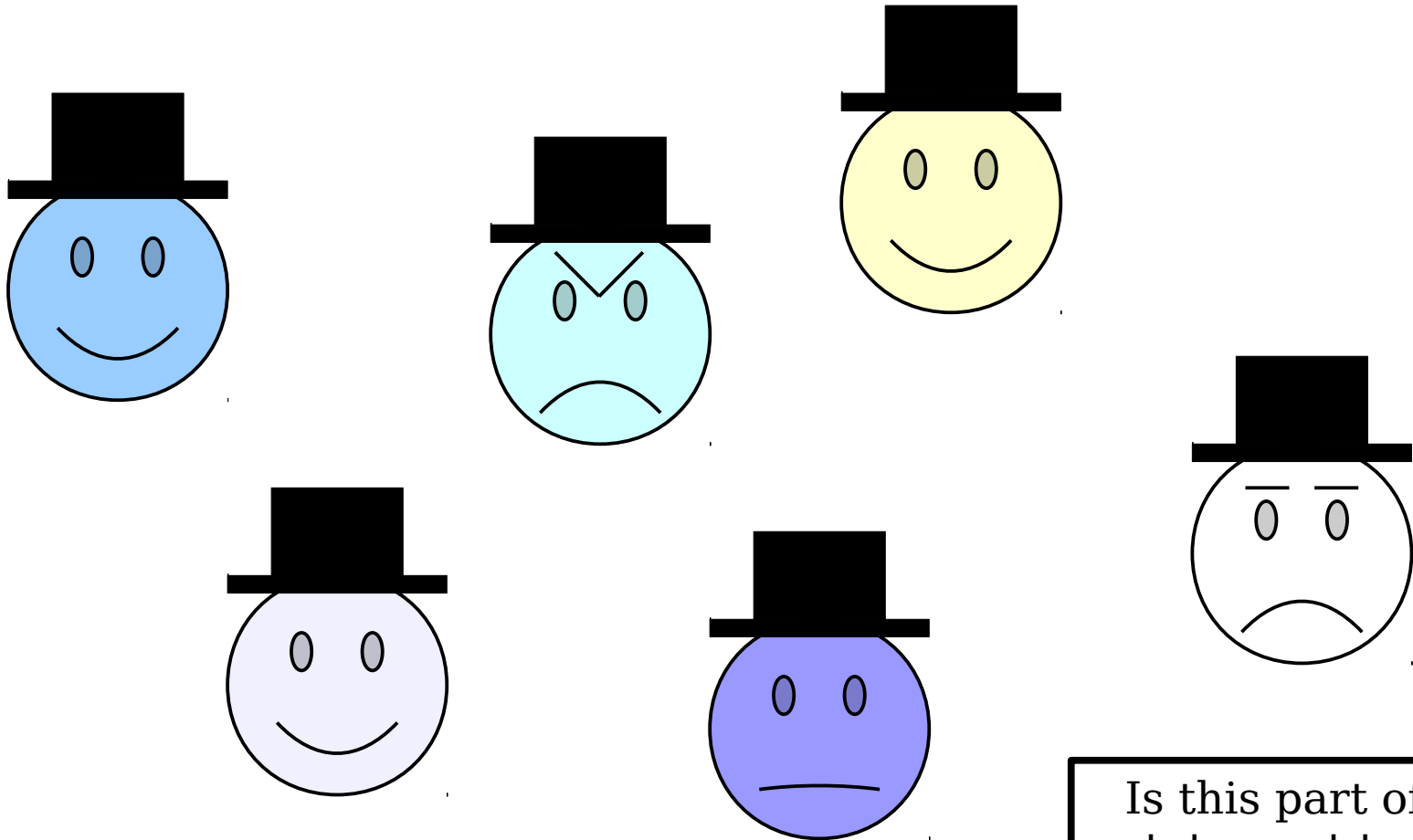
The Universal Quantifier



Answer at
<https://cs103.stanford.edu/pollev>

$$(\forall x. \textit{Smiling}(x)) \rightarrow (\forall y. \textit{WearingHat}(y))$$

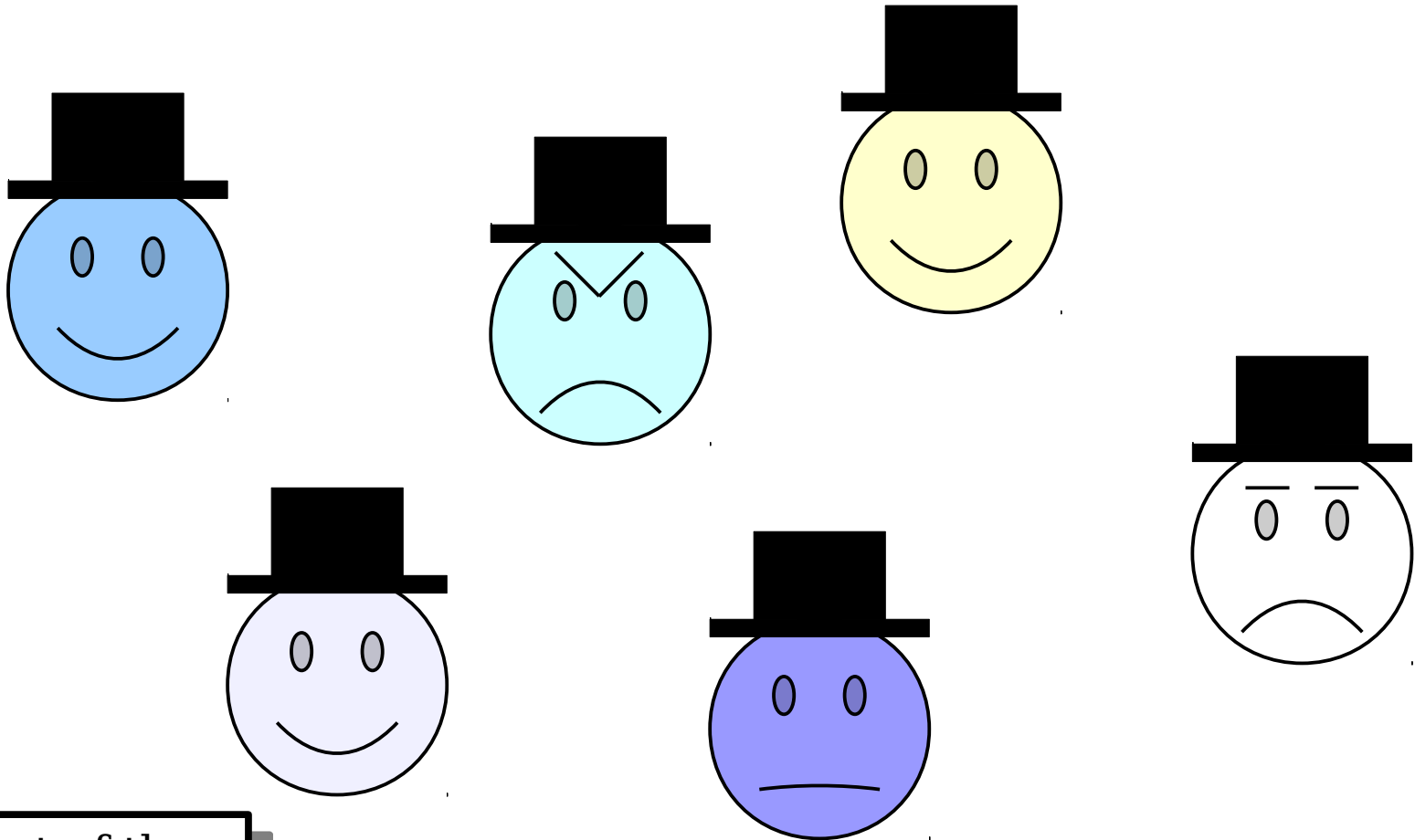
The Universal Quantifier



Is this part of the statement true or false?

$(\forall x. \textit{Smiling}(x)) \rightarrow (\forall y. \textit{WearingHat}(y))$

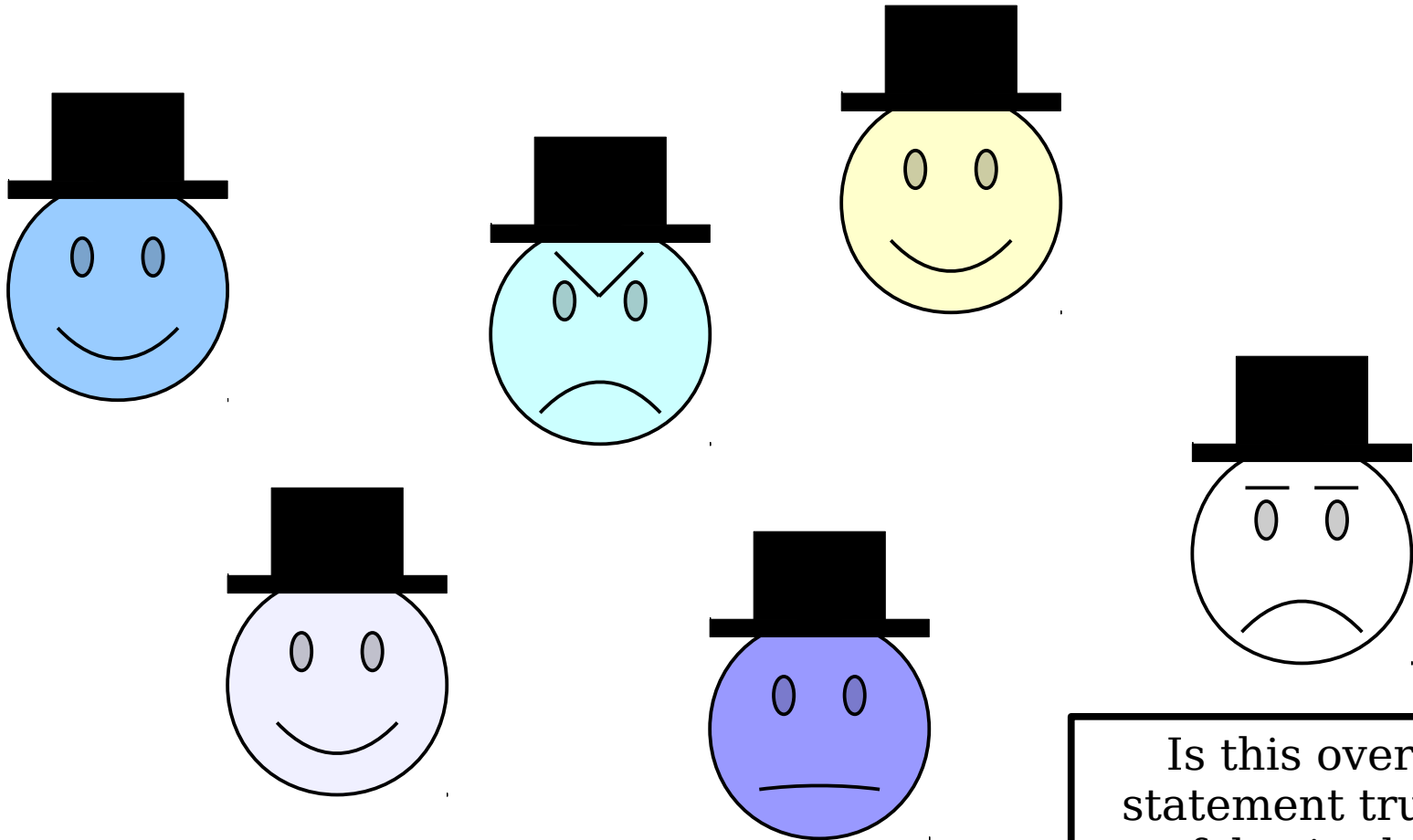
The Universal Quantifier



Is this part of the
statement true or
false?

$$(\forall x. \textit{Smiling}(x)) \rightarrow (\forall y. \textit{WearingHat}(y))$$

The Universal Quantifier



Is this overall
statement true or
false in this
scenario?

$$(\forall x. \textit{Smiling}(x)) \rightarrow (\forall y. \textit{WearingHat}(y))$$

Fun with Edge Cases

Universally-quantified statements are said to be ***vacuously true*** in empty worlds.

$\forall x. \text{Smiling}(x)$

Translating into First-Order Logic

Translating Into Logic

- First-order logic is an excellent tool for manipulating definitions and theorems to learn more about them.
- Need to take a negation? Translate your statement into FOL, negate it, then translate it back.
- Want to prove something by contrapositive? Translate your implication into FOL, take the contrapositive, then translate it back.

Translating Into Logic

- When translating from English into first-order logic, we recommend that you
think of first-order logic as a mathematical programming language.
- Your goal is to learn how to combine basic concepts (quantifiers, connectives, etc.) together in ways that say what you mean.

Using the predicates

- *Smiling*(x), which states that x is smiling, and
- *WearingHat*(x), which states that x is wearing a hat,

write a formula in first-order logic that says

some smiling person wears a hat.

How would you represent this in first-order logic?

Answer at

<https://cs103.stanford.edu/pollev>

Using the predicates

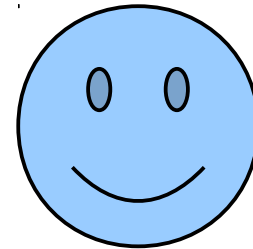
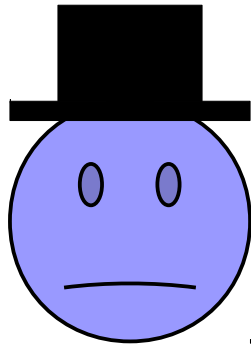
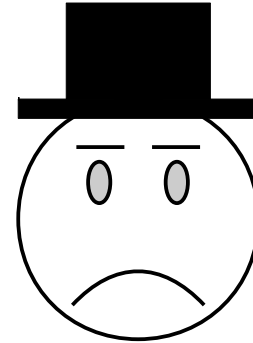
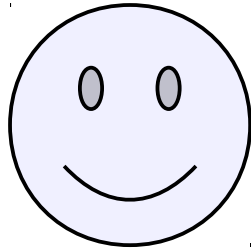
- $Smiling(x)$, which states that x is smiling, and
- $WearingHat(x)$, which states that x is wearing a hat,

write a formula in first-order logic that says

some smiling person wears a hat.

Which of the following are correct translations?

- (A) ~~$\exists x. Smiling(Person(x))$~~
- (B) ~~$\exists x. (Smiling(x) = WearingHat(x))$~~
- (C) $\exists x. (Smiling(x) \wedge WearingHat(x))$
- (D) $\exists x. (Smiling(x) \rightarrow WearingHat(x))$



“Some smiling person wears a hat.” ***False***

$\exists x. (Smiling(x) \wedge WearingHat(x))$ ***False***

~~$\exists x. (Smiling(x) \rightarrow WearingHat(x))$~~ ***True***

“Some P is a Q ”

translates as

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

Useful Intuition:

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

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

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

Using the predicates

- *Smiling*(x), which states that x is smiling, and
- *WearingHat*(x), which states that x is wearing a hat,

write a sentence in first-order logic that says

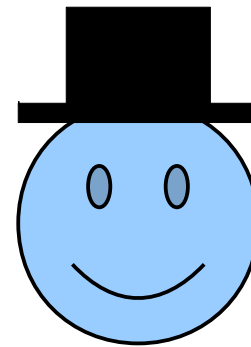
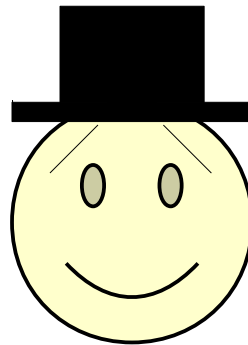
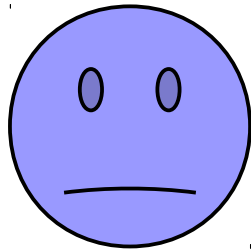
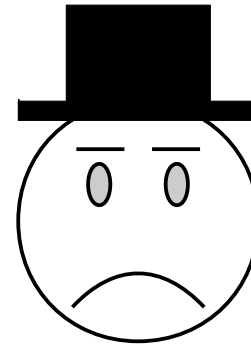
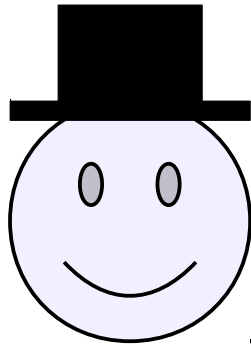
every smiling person wears a hat.

Which of the following are correct translations?

- (A) $\forall x. (\text{Smiling}(x) \wedge \text{WearingHat}(x))$
- (B) $\forall x. (\text{Smiling}(x) \rightarrow \text{WearingHat}(x))$

Answer at

<https://cs103.stanford.edu/pollev>



“Every smiling person wears a hat.” **True**

~~$\forall x. (Smiling(x) \wedge WearingHat(x))$~~ **False**

$\forall x. (Smiling(x) \rightarrow WearingHat(x))$ **True**

“All P 's are Q 's”

translates as

$\forall x. (P(x) \rightarrow Q(x))$

Useful Intuition:

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

$$\forall x. (P(x) \rightarrow Q(x))$$

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

Good Pairings

- The $\forall$ quantifier *usually* is paired with $\rightarrow$.

$$\forall x. (P(x) \rightarrow Q(x))$$

- The $\exists$ quantifier *usually* is paired with $\wedge$.

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

- 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 $\wedge$ connective prevents the statement from being *true* when speaking about some object you don't care about.

Next Time

- ***First-Order Translations***
 - How do we translate from English into first-order logic?
- ***Quantifier Orderings***
 - How do you select the order of quantifiers in first-order logic formulas?
- ***Negating Formulas***
 - How do you mechanically determine the negation of a first-order formula?
- ***Expressing Uniqueness***
 - How do we say there's just one object of a certain type?