Assignment 1
   Graded

Assignment 2
   Not graded

Assignment 3
   Due Thursday
Where are we?
A **propositional variable** is a variable that is either true or false.

The **propositional connectives** are as follows:

- Negation: $\neg \varphi$
- Conjunction: $\varphi \land \psi$
- Disjunction: $\varphi \lor \psi$
- Implication: $\varphi \rightarrow \psi$
- Biconditional: $\varphi \leftrightarrow \psi$
- True: $\top$
- False: $\bot$
Two propositional formulas $\varphi$ and $\psi$ are called \textit{equivalent} if they have the same truth tables.

We denote this by writing $\varphi \equiv \psi$.

Some examples:

$$
\neg(\varphi \land \psi) \equiv \neg \varphi \lor \neg \psi
$$

$$
\neg(\varphi \lor \psi) \equiv \neg \varphi \land \neg \psi
$$

$$
\neg \varphi \lor \psi \equiv \varphi \Rightarrow \psi
$$

$$
\varphi \land \neg \psi \equiv \neg(\varphi \Rightarrow \psi)
$$
Take out a sheet of paper!
What’s the truth table for the \( \Rightarrow \) connective?
What’s the negation of $p \implies q$?
First-order logic
Concept-Script

A Formal Language of Pure Thought on the Pattern of Arithmetic
First-order logic is a logical system for reasoning about properties of objects.

It augments the 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.
Likes(You, Music) \land Likes(You, Plays) \\
\implies Likes(You, MusicalTheater)

Learns(You, History) \lor \\
ForeverRepeats(You, History)
Likes(You, Music) \land Likes(You, Plays)
\Rightarrow Likes(You, MusicalTheater)

Learns(You, History) \lor
ForeverRepeats(You, History)

These are constant symbols. They refer to objects, not propositions.
Likes(You, Music) ∧ Likes(You, Plays)
⇒ Likes(You, MusicalTheater)

Learns(You, History) ∨
ForeverRepeats(You, History)

These are *predicates*. They take objects as arguments and evaluate to true or false.
Likes(You, Music) \land Likes(You, Plays)
\Rightarrow Likes(You, MusicalTheater)

Learns(You, History) \lor
ForeverRepeats(You, History)

What remain are traditional propositional connectives. Because each predicate evaluates to true or false, we can connect the truth values of predicates using normal propositional connectives.
To reason about objects, first-order logic uses **predicates**, e.g.,

- Cute(Quokka)
- Argue(Democrats, Republicans)

Applying a predicate to arguments produces a proposition, which is either true or false.
Each predicates can takes a fixed number of arguments, called its *arity*.

So, in first-order logic, you can’t have both

- Eat(Garfield, Lasagna)
- Eat(Garfield, Lasagna, Home)

Since they use the same predicate, Eat, with different numbers of arguments.
Sentences in first-order logic can be constructed from predicates applied to objects:

\[
\text{Cute}(A) \Rightarrow \text{Bunny}(A) \lor \text{Kitty}(A) \lor \text{Puppy}(A)
\]

\[
\text{Succeeds}(\text{You}) \iff \text{Practices}(\text{You})
\]

\[
x < 8 \Rightarrow x < 137
\]

The < sign is just another predicate. Binary predicates are sometimes written in infix notation this way.

Numbers aren’t “built in” to FOL. They’re constant symbols just like “You” or “A” above.
First-order logic is equipped with the special predicate $=\,$ that says whether two objects are equal to one another.

$\text{MorningStar} = \text{EveningStar}$

$\text{TomMarvoloRiddle} = \text{LordVoldemort}$
Equality is a part of FOL, just as $\Rightarrow$ and $\neg$ are. For notational simplicity, define $\neq$ as

$$x \neq y \equiv \neg(x = y).$$

Equality can only be applied to objects; to see if propositions are equal, use $\leftrightarrow$. 
favoriteMovieOf(You) ≠ favoriteMovieOf(Date) ∧
starOf(favoriteMovieOf(You)) = starOf(favoriteMovieOf(Date))
Constants

\[\text{favoriteMovieOf}(\text{You}) \neq \text{favoriteMovieOf}(\text{Date}) \land \text{starOf(favoriteMovieOf(You))} = \text{starOf(favoriteMovieOf(Date))}\]
These are functions. Functions take objects as input and produce objects as output.
favoriteMovieOf(You) ≠ favoriteMovieOf(Date) \∧ 
starOf(favoriteMovieOf(You)) = starOf(favoriteMovieOf(Date))

Predicates
favoriteMovieOf(You) \neq favoriteMovieOf(Date) \land 
starOf(favoriteMovieOf(You)) = starOf(favoriteMovieOf(Date))
First-order logic allows functions, which return objects associated with other functions.

Examples:

- `colorOf(Money)`
- `medianOf(X, Y, Z)`
- `X + Y`

As with predicates, functions can have any fixed arity, but they always return a single value.

Functions evaluate to objects, not propositions.
When working in first-order logic, be careful to keep objects (things) and propositions (true or false) separate.

You cannot apply connectives to objects:

$$\text{Venus } \Rightarrow \text{ TheSun}$$

You cannot apply functions to propositions:

$$\text{starOf} (\text{IsRed}(\text{Sun}) \land \text{IsGreen}(\text{Mars}))$$
## Type-checking table

<table>
<thead>
<tr>
<th></th>
<th><strong>Operate on</strong></th>
<th><strong>Produce</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Connectives</strong></td>
<td>propositions</td>
<td>a proposition</td>
</tr>
<tr>
<td>like: ( \Rightarrow ) and ( \wedge )</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Predicates</strong></td>
<td>objects</td>
<td>a proposition</td>
</tr>
<tr>
<td>like: = and Loves</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Functions</strong></td>
<td>objects</td>
<td>an object</td>
</tr>
<tr>
<td>like: ageOf and length</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
One last (major) change
Some muggle is intelligent.
Some muggle is intelligent.

\( \exists m . \ (Muggle(m) \land Intelligent(m)) \)
Some muggle is intelligent.

\[ \exists m . \ (\text{Muggle}(m) \land \text{Intelligent}(m)) \]

\[ \exists \] is the existential quantifier and says “for some choice of \( m \), the following is true”. 
The existential quantifier

A statement of the form

\[ \exists x . \varphi \]

is true if, for some choice of \( x \), the statement \( \varphi \) is true when that \( x \) is plugged into it.

Examples:

\[ \exists x . (\text{Even}(x) \land \text{Prime}(x)) \]
\[ \exists x . (\text{TallerThan}(x, \text{Me}) \land \text{LighterThan}(x, \text{Me})) \]
\[ (\exists w . \text{Will}(w)) \Rightarrow (\exists x . \text{Way}(x)) \]
The existential quantifier

∃x . Smiling(x)
The existential quantifier

∃x . Smiling(x)
The existential quantifier

$\exists x . \text{Smiling}(x)$
The existential quantifier

\( \exists x . \text{Smiling}(x) \)
The existential quantifier

$\exists x . \text{Smiling}(x)$
The existential quantifier

∃x . Smiling(x)

Since Smiling(x) is true for some choice of x, this statement evaluates to true.
The existential quantifier

\[ \exists x \cdot \text{Smiling}(x) \]
The existential quantifier

∃x. Smiling(x)
The existential quantifier

$\exists x \cdot \text{Smiling}(x)$
The existential quantifier

\[ \exists x . \text{Smiling}(x) \]
The existential quantifier

∃x . Smiling(x)
The existential quantifier

∃x . Smiling(x)
The existential quantifier

\[ \exists x . \text{Smiling}(x) \]

Since \( \text{Smiling}(x) \) isn’t true for any choice of \( x \), this statement evaluates to false.
The existential quantifier

(∃x . Smiling(x)) ⇒ (∃y . WearingHat(y))
The existential quantifier

Is this part of the statement true or false?

\((\exists x . \text{Smiling}(x)) \Rightarrow (\exists y . \text{WearingHat}(y))\)
The existential quantifier

Is this part of the statement true or false?

\( T \Rightarrow (\exists y . \text{WearingHat}(y)) \)
The existential quantifier

Is the overall statement true or false?

\[ \top \Rightarrow \bot \]
The existential quantifier

Is the overall statement true or false?
The existential quantifier

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

$\exists x. \text{Smiling}(x)$
Variables and quantifiers

Each quantifier has two parts:

the variable that is being introduced, and

the statement that’s being quantified.

The variable introduced is scoped to just the statement being quantified.

$$(\exists x . \text{Loves(You, } x)) \land (\exists y . \text{Loves(y, You)})$$
Variables and quantifiers

Each quantifier has two parts:

the variable that is being introduced, and

the statement that’s being quantified.

The variable introduced is scoped to just the statement being quantified.

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

The variable $x$ is only defined here

The variable $y$ is only defined here
Variables and quantifiers

Each quantifier has two parts:

the variable that is being introduced, and

the statement that’s being quantified.

The variable introduced is scoped to just the statement being quantified.

\[(\exists x . \text{Loves}(\text{You}, x)) \land (\exists x . \text{Loves}(x, \text{You}))\]

\begin{align*}
&\text{The variable } x \text{ is only defined here} & & \text{A different variable named } x \text{ is only defined here}
\end{align*}
Scope in first-order logic is much like in programming:

\[(\exists x . \text{Loves}(\text{You, } x)) \land (\exists x . \text{Loves}(x, \text{You}))\]

corresponds to separate environments:

(let [(x "Sarah")]
 (printf "~A is great~%" x))
(let [(x "Jorge")]
 (printf "~A is also great~%" x))
For any natural number $n$, $n$ is even if and only if $n^2$ is even.
For any natural number $n$, $n$ is even if and only if $n^2$ is even.

\[ \forall n . (n \in \mathbb{N} \Rightarrow (\text{Even}(n) \leftrightarrow \text{Even}(n^2))) \]
For any natural number $n$, $n$ is even if and only if $n^2$ is even.

$$\forall n . \ (n \in \mathbb{N} \Rightarrow (\text{Even}(n) \iff \text{Even}(n^2)))$$

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

A statement of the form

$$\forall x . \varphi$$

is true if, for every choice of $x$, the statement $\varphi$ is true when that $x$ is plugged into it.

Examples:

$$\forall k . (Kitten(k) \Rightarrow Cute(k))$$

Richest(BernardArnault) \Rightarrow

$$\forall x . (x \neq BernardArnault \Rightarrow RicherThan(BernardArnault, x))$$
The universal quantifier

∀x . Smiling(x)
The universal quantifier

∀x . Smiling(x)
The universal quantifier

∀x. Smiling(x)
The universal quantifier

\( \forall x . \text{Smiling}(x) \)
The universal quantifier

∀x. Smiling(x)
The universal quantifier

∀x . Smiling(x)
The universal quantifier

\[ \forall x . \text{Smiling}(x) \]

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

∀x . Smiling(x)
The universal quantifier

∀x . Smiling(x)
The universal quantifier

∀x . Smiling(x)
The universal quantifier

$$\forall x. \text{Smiling}(x)$$
The universal quantifier

\[ \forall x . \text{Smiling}(x) \]

Since \( \text{Smiling}(x) \) is false for this choice of \( x \), this statement evaluates to false.
The universal quantifier

$$(\forall x \cdot \text{Smiling}(x)) \Rightarrow (\forall y \cdot \text{WearingHat}(y))$$
The universal quantifier

(∀x . Smiling(x)) ⇒ (∀y . WearingHat(y))

Is this part of the statement true or false?
The universal quantifier

Is this part of the statement true or false?

\( (\forall x . \text{Smiling}(x)) \Rightarrow T \)
The universal quantifier

\[ \forall x. \text{Smiling}(x) \]

\[ \forall y. \text{WearingHat}(y) \]
The universal quantifier

Is this overall statement true or false in this scenario?

⊥ \Rightarrow T
The universal quantifier

Is this overall statement true or false in this scenario?

T
The universal quantifier

\[ \forall x . \text{Smiling}(x) \]

Universally quantified statements are vacuously true in an empty world.
Translating into first-order 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 turn it back.

Want to prove something by contrapositive? Translate your implication into FOL, take the contrapositive, then translate back.
When translating from English into first-order logic, think of FOL 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

\[ \text{Happy}(x), \text{which states that } x \text{ is happy, and} \]
\[ \text{WearingHat}(x), \text{which states that } x \text{ is wearing a hat,} \]

write a sentence in first-order logic that says

\[ \text{Some happy person wears a hat.} \]
“Some happy person wears a hat.”

$$\exists x . \ (\text{Happy}(x) \land \text{WearingHat}(x))$$

$$\exists x . \ (\text{Happy}(x) \implies \text{WearingHat}(x))$$
“Some happy person wears a hat.”

\[ \exists x . \ (\text{Happy}(x) \land \text{WearingHat}(x)) \]

\[ \exists x . \ (\text{Happy}(x) \implies \text{WearingHat}(x)) \]
Some happy person wears a hat.

\[ \exists x . (\text{Happy}(x) \land \text{WearingHat}(x)) \]

\[ \exists x . (\text{Happy}(x) \Rightarrow \text{WearingHat}(x)) \]
∃x. (Happy(x) ∧ WearingHat(x))

∃x. (Happy(x) ⇒ WearingHat(x))
“Some happy person wears a hat.”

\[ \exists x . (\text{Happy}(x) \land \text{WearingHat}(x)) \]

\[ \exists x . (\text{Happy}(x) \Rightarrow \text{WearingHat}(x)) \]
“Some happy person wears a hat.”

$\exists x . (\text{Happy}(x) \land \text{WearingHat}(x))$

$\exists x . (\text{Happy}(x) \Rightarrow \text{WearingHat}(x))$
“Some happy person wears a hat.”

$\exists x . (\text{Happy}(x) \land \text{WearingHat}(x))$

$\exists x . (\text{Happy}(x) \Rightarrow \text{WearingHat}(x))$
"Some happy person wears a hat."

\[ \exists x . (\text{Happy}(x) \land \text{WearingHat}(x)) \]

\[ \exists x . (\text{Happy}(x) \implies \text{WearingHat}(x)) \]
Intuitively, these people should be irrelevant.

"Some happy person wears a hat."

\[ \exists x. \ (\text{Happy}(x) \land \text{WearingHat}(x)) \]

\[ \exists x. \ (\text{Happy}(x) \Rightarrow \text{WearingHat}(x)) \]
“Some happy person wears a hat.”

∃x . (Happy(x) ∧ WearingHat(x))

∃x . (Happy(x) ⇒ WearingHat(x))
∃x. (Happy(x) ∧ WearingHat(x))

∃x. (Happy(x) ⇒ WearingHat(x))
"Some happy person wears a hat."

$$\exists x . (\text{Happy}(x) \land \text{WearingHat}(x))$$

$$\exists x . (\text{Happy}(x) \Rightarrow \text{WearingHat}(x))$$
∃x . (Happy(x) ⇒ WearingHat(x))
“Some $P$ is a $Q$” translates as

$$\exists x \cdot (P(x) \land Q(x))$$
∃x . (P(x) ∧ Q(x))

Useful intuition: Existentially quantified statements are false unless there’s a positive example
Using the predicates

\[ \text{Happy}(x), \text{which states that } x \text{ is happy, and} \]
\[ \text{WearingHat}(x), \text{which states that } x \text{ is wearing a hat,} \]

write a sentence in first-order logic that says

\textit{Every happy person wears a hat.}
“Every happy person wears a hat.”

\[ \forall x . (\text{Happy}(x) \land \text{WearingHat}(x)) \]

\[ \forall x . (\text{Happy}(x) \Rightarrow \text{WearingHat}(x)) \]
“Every happy person wears a hat.”

\[ \forall x . (\text{Happy}(x) \land \text{WearingHat}(x)) \]

\[ \forall x . (\text{Happy}(x) \Rightarrow \text{WearingHat}(x)) \]
“Every happy person wears a hat.”

\[ \forall x . (\text{Happy}(x) \land \text{WearingHat}(x)) \]

\[ \forall x . (\text{Happy}(x) \implies \text{WearingHat}(x)) \]
“Every happy person wears a hat.”

∀x . (Happy(x) ∧ WearingHat(x))

∀x . (Happy(x) ⇒ WearingHat(x))
“Every happy person wears a hat.”

\[ \forall x . \ (\text{Happy}(x) \land \text{WearingHat}(x)) \]

\[ \forall x . \ (\text{Happy}(x) \Rightarrow \text{WearingHat}(x)) \]
“Every happy person wears a hat.”

\[ \forall x . (\text{Happy}(x) \land \text{WearingHat}(x)) \]

\[ \forall x . (\text{Happy}(x) \implies \text{WearingHat}(x)) \]
"Every happy person wears a hat."

∀x . (Happy(x) ∧ WearingHat(x))

∀x . (Happy(x) → WearingHat(x))
“Every happy person wears a hat.”

\[
\forall x . (\text{Happy}(x) \land \text{WearingHat}(x))
\]

\[
\forall x . (\text{Happy}(x) \implies \text{WearingHat}(x))
\]
“All Ps are Qs” translates as

\[ \forall x . (P(x) \implies Q(x)) \]
\[ \forall x \ . \ (P(x) \implies Q(x)) \]

**Useful intuition:** Universally quantified statements are true unless there's a counterexample.
Good pairings

The $\forall$ quantifier *usually* is paired with $\Rightarrow$.

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

The $\exists$ quantifier usually is paired with $\land$.

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

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 $\land$ connective prevents the statement from being *true* when speaking about some object you don’t care about.
Acknowledgments

This lecture incorporates material from:

Keith Schwarz