Predicate Logic
Propositional logic dealt with logical forms of compound propositions. It worked well with relationships like not, and, or, if/then.

We would like to have a way to talk about *individuals* (also called *objects*) and in addition to talk about *some* object, and *all* objects, without enumerating all objects in a set.

This requires extensions to Propositional Logic.
Some example statements:

*Not all birds can fly.*

*Every student is younger than some instructor.*

These refer to things: birds, students, instructors. They also refer to properties of things, either as individuals (ability to fly) or in combination (relative age).

We would like to make such statements in our logic and to combine them with the connectives of propositional logic.

Statements like those above require a context, or “world”, to be meaningful.
Informal Introduction to Predicate Logic

We shall briefly discuss some “worlds” for predicate logic.

- Arithmetic (the inspiration for formal predicate logic).
- The “blocks world” (a classical AI example).
- Graphs (a ubiquitous concept in CS).
- Databases (the “applied CS” version of predicate logic).

In each, we shall see the following ingredients.

- Domains. The set of objects; also called the “universe”.
- Constants. Objects with specific names.
- Predicates (aka Relations). Properties of objects, alone or in combination.
- Functions. Association of objects to others.
Arithmetic, Etc.

In arithmetic, we have several possible domains.

- \( \mathbb{N} \): The natural numbers
- \( \mathbb{Z} \): The integers
- \( \mathbb{Q} \): The rationals
- Integers modulo \( n \)
- ...

Over these, we have *predicates* (or *relations*):

- \( < \): Less than
- “Has a square root”
- ...

and also functions: +, \( \times \), etc.

A sample formula: \( (\forall x (\exists y (y < x))) \).
A Classic AI Example: Blocks Worlds

A “blocks world” consists of a set of blocks, and a table.

- Each block may be on the table, or on one of the other blocks.
- Each block may have a colour.

In the picture, there are three blocks.

Two of them are blue (vertical stripes) and one is red (diagonal stripes).

Set of objects: \{B1, B2, B3\}.

We can describe the world with predicates: \textit{On}, \textit{OnTable}, \textit{Red}, \textit{Blue}, ....
Describing a Blocks World

The domain \( \{B_1, B_2, B_3\} \) is a finite set.

Therefore, we can list all of the properties, in various ways:

- \( OnTable(B_1), OnTable(B_2), (\neg OnTable(B_3)) \).

- \( On: \)

<table>
<thead>
<tr>
<th></th>
<th>B1</th>
<th>B2</th>
<th>B3</th>
</tr>
</thead>
<tbody>
<tr>
<td>B1</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>B2</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>B3</td>
<td>T</td>
<td>F</td>
<td>F</td>
</tr>
</tbody>
</table>

- The set \( \{ b \mid Blue(b) \} \) of blue blocks is \( \{B_2, B_3\} \).
Properties in the Blocks World

Some properties are fundamental to the world.

“No box is on itself” \( (\forall x (\neg \text{On}(x, x))) \).

“A box on the table is not on any box”:

\[
\left( \forall x \left( \text{OnTable}(x) \rightarrow (\neg (\exists y \ \text{On}(x, y))) \right) \right).
\]

Some properties depend on the situation.

“Every red box has a box on it”: \( (\forall x \left( \text{Red}(x) \rightarrow (\exists y \ \text{On}(y, x)) \right)) \).

“Some box is on a box that is on the table”:

\[
\left( \exists x \left( \exists y \left( \text{On}(x, y) \land \text{OnTable}(y) \right) \right) \right).
\]
A graph is a binary predicate.

A finite graph:

\[ \begin{align*}
\text{An infinite graph:} & \\
(\text{The set } \{ \langle x, y \rangle \mid y = 1 + x/2 \}).
\end{align*} \]

A graph is undirected if the predicate is symmetric; i.e., the formula

\[ \left( \forall x \left( \forall y \left( E(x, y) \rightarrow E(y, x) \right) \right) \right) \]

holds.
A *relational database* is a listing of one or more predicates (relations).

Example:

- *Person*: The people (or their names).
- *NumberOf*: An association between people and their phone numbers.

Here the domain contains both people and phone numbers — the objects about which we have predicates.

A sample statement: “Somebody has no phone number.”

\[
(\exists x \left( Person(x) \land (\neg(\exists y \left( NumberOf(x, y)\right))))) \right) .
\]
A Conundrum

Consider the statement, “Only people have phone numbers.” How shall we represent it as a logical formula?

“Whenever $x$ and $y$ satisfy $\text{NumberOf}(x, y)$, then $x$ is a person.”

\[
(\forall x (\forall y (\text{NumberOf}(x, y) \rightarrow \text{Person}(x))))
\]

“Whenever $x$ has some phone number $y$, then $x$ is a person.”

\[
(\forall x ((\exists y \text{NumberOf}(x, y)) \rightarrow \text{Person}(x)))
\]

These two formulas are respectively equivalent to

\[
(\forall x (\forall y (((\neg \text{NumberOf}(x, y)) \lor \text{Person}(x)))))
\]

and to

\[
(\forall x (\left(\neg (\exists y \text{NumberOf}(x, y))\right) \lor \text{Person}(x)))
\]

Are these two formulas equivalent?
We now turn to general definitions.

A *domain* is a non-empty set. In principle, any non-empty set can be a domain: the natural numbers, people now alive, \{T, F\}, etc.

A *constant symbol* refers to an object in the domain; e.g., 0, \(B_1\), Justin Trudeau, etc.
A \textit{predicate}, or \textit{relation}, represents a property that an individual, or collection of individuals, may (or may not) have. In English, we might express a predicate as

“\text{_______ is a student}”.

In symbolic logic, we write “$S(x)$” to mean “$x$ has property $S$”.

For example, if $S$ is the property of being a student, then “Alex is a student” becomes “$S(\text{Alex})$”.

Similarly, we might use $I(\text{Sam})$ for “Sam is an instructor”

and $Y(\text{Alex, Sam})$ for “Alex is younger than Sam”.

Representing Predicates

Mathematically, we represent a predicate by the set of all things that have the property. If $S$ is the set of all students, then $x \in S$ means $x$ is a student. The only restriction on a predicate is that it must be a subset of the domain.

A $k$-ary predicate (relation) is a set of $k$-tuples of domain elements. For example, the binary predicate less-than, over a domain $D$, is represented by the set

$$\left\{ \langle x, y \rangle \in D^2 \mid x < y \right\}.$$

As another example, the $On$ predicate in the sample blocks world has just one pair: $\{\langle B3, B1 \rangle\}$.

(In a “relational database”, the listing of such a set is called a “table”.)
Variables make statements more expressive.
You may think of a variable as a “place holder”, or “blank”, that can be replaced by a concrete object.

Alternatively, a variable is a name without a fixed referent. Which object the name refers to can vary from time to time.

A variable lets us refer to an object, without specifying—perhaps without even knowing—which particular object it is. Thus we can express a predicate “in the abstract”.

\[
S(x): \quad x \text{ is a student} \\
I(x): \quad x \text{ is an instructor} \\
Y(x, y): \quad x \text{ is younger than } y
\]
In general, we use variables that range over the domain to make general statements, such as

\[ x^2 \geq 0 \]  

and in expressing conditions which individuals may or may not satisfy, such as

\[ x + x = x \times x \]  

This latter condition is satisfied by only two numbers: 0 and 2.

The meaning of such an expression will depend on the domain. For example, the formula \( x^2 < x \) is always false over the domain of integers, but not over the domain of rational numbers.
What about “**Every** student \(x\) is younger than **some** professor \(y\)”?

In math-speak, we say “for all” to express “every” and “there exists” to express “some.” A familiar(?) example from calculus:

> For all \(\varepsilon > 0\), there exists \(\delta > 0\) such that
> for all \(y\), if \(|x - y| < \delta\) then \(|f(x) - f(y)| < \varepsilon\).

“For all” is denoted by ‘\(\forall\)’, the *universal quantifier* symbol, and “there exists” is denoted by ‘\(\exists\)’, the *existential quantifier* symbol.

In Predicate Logic, the above comes out as the formula

\[
\left( \forall \varepsilon \left( (\varepsilon > 0) \rightarrow \exists \delta \left( (\delta > 0) \land \forall y \left( (|x - y| < \delta) \rightarrow (|f(x) - f(y)| < \varepsilon) \right) \right) \right)
\]
Quantifiers: Examples

Quantifiers require a variable: \( \forall x \) (for all \( x \)) or \( \exists z \) (there exists \( z \)).

For example, the statement “Not all birds can fly” can be written as

\[
\neg (\forall x \ (B(x) \rightarrow F(x)))
\]

“Every student is younger than some instructor” can become

\[
\forall x \ (S(x) \rightarrow (\exists y \ (I(y) \land Y(x, y))))
\]

Or should that be

\[
(\exists y \ (I(y) \land (\forall x \ (S(x) \rightarrow Y(x, y)))))
\]

These two formulas are NOT equivalent!
Syntax of Predicate Logic
The Language of Predicate Logic

The seven kinds of symbols:

1. **Constant symbols.** Usually $c, d, c_1, c_2, \ldots, d_1, d_2 \ldots$
2. **Variables.** Usually $x, y, z, \ldots x_1, x_2, \ldots, y_1, y_2 \ldots$
3. **Function symbols.** Usually $f, g, h, \ldots f_1, f_2, \ldots, g_1, g_2, \ldots$
4. **Predicate symbols.** $P, Q, \ldots P_1, P_2, \ldots, Q_1, Q_2, \ldots$
5. **Connectives:** $\neg, \land, \lor, \rightarrow, \text{ and } \leftrightarrow$
6. **Quantifiers:** $\forall$ and $\exists$
7. **Punctuation:** ‘(’, ‘)’, and ‘,’

Function symbols and predicate symbols have an assigned *arity*—the number of arguments required.

The last three kinds of symbols—connectives, quantifiers, and punctuation—will have their meaning fixed by the syntax and semantics.

Constants, variables, functions and predicate symbols are not restricted. They may be assigned any meaning, consistent with their kind and arity.
In Predicate Logic, we need to consider two kinds of expressions:

- those that can have a truth value, called *formulas*, and
- those that refer to an object of the domain, called *terms*.

We start with terms.

**Definition.** The set of terms is defined inductively as follows.

1. Each constant symbol is a term, and each variable is a term. Such terms are called *atomic* terms.

2. If \( t_1, \ldots, t_n \) are terms and \( f \) is an \( n \)-ary function symbol, then \( f(t_1, \ldots, t_n) \) is a term. If \( n = 2 \) (i.e. if \( f \) is a binary function symbol), then we may write \( (t_1 f t_2) \) instead of \( f(t_1, t_2) \).

3. Nothing else is a term.
Examples of Terms

**Example 1.** If 0 is a constant symbol, $x$ and $y$ are variables, and $s^{(1)}$ and $+^{(2)}$ are function symbols, then $0$, $x$, and $y$ are terms, as are $s(0)$ and $+(x, s(y))$.

The expressions $s(x, y)$ and $s + x$ are not terms.

**Example 2.** Suppose $f$ is a unary function symbol, $g$ is a binary function symbol, and $a$ is a constant symbol.

Then $g(f(a), a)$ and $f(g(a, f(a)))$ are terms.

The expressions $g(a)$ and $f(f(a), a)$ are not terms.
As in propositional logic, a formula represents a proposition (a true/false statement). The predicate symbols produce propositions.

**Definition:** An *atomic formula* (or atom) is an expression of the form

\[ P(t_1, \ldots, t_n) \]

where \( P \) is an \( n \)-ary predicate symbol and each \( t_i \) is a term \((1 \leq i \leq n)\). If \( P \) has arity 2, the atom \( P(t_1, t_2) \) may alternatively be written \((t_1 \; P \; t_2)\).
We define the set of well-formed formulas of Predicate logic inductively as follows.

1. An atomic formula is a formula.
2. If \( \alpha \) is a formula, then \( (\neg \alpha) \) is a formula.
3. If \( \alpha \) and \( \beta \) are formulas, and \( \star \) is a binary connective symbol, then \( (\alpha \star \beta) \) is a formula.
4. If \( \alpha \) is a formula and \( x \) is a variable, then each of \( (\forall x \ \alpha) \) and \( (\exists x \ \alpha) \) is a formula.
5. Nothing else is a formula.

In case 4, the formula \( \alpha \) is called the scope of the quantifier. The quantifier keeps the same scope if it is included in a larger formula.
Parse Trees

Parse trees for Predicate Logic formulas are similar to parse trees for propositional formulas.

- A function $f(t_1, t_2, \ldots, t_n)$ has a node labelled $f$ with a sub-tree for each of the terms $t_1, t_2, \ldots, t_n$.
- A predicate $P(t_1, t_2, \ldots, t_n)$ has a node labelled $P$ with a sub-tree for each of the terms $t_1, t_2, \ldots, t_n$.
- Quantifiers $\forall x$ and $\exists y$ form nodes is the same way as negation (i.e., only one sub-tree).
Examples: Parse trees

Example: \((\forall x ((P(x) \rightarrow Q(x)) \land S(x, y)))).\)
Examples: Parse trees

Example: $\forall x \left( \left( P(x) \rightarrow Q(x) \right) \land S(x, y) \right)$.

Example: $\forall x \left( F(b) \rightarrow \exists y \left( \forall z \left( G(y, z) \lor H(u, x, y) \right) \right) \right)$.
Examples: Parse trees

Example: \((\forall x ((P(x) \rightarrow Q(x)) \land S(x, y)))\).

Example: \((\forall x (F(b) \rightarrow (\exists y (\forall z (G(y, z) \lor H(u, x, y)))))\))

Ordinarily, one would omit many of the parentheses in the second formula, and write simply

\[ \forall x (F(b) \rightarrow \exists y \forall z (G(y, z) \lor H(u, x, y))) \]
We shall discuss

- “bound” and “free” variables,
- substitution of terms for variables.
Free and Bound Variables

Recall: the *scope* of a quantifier in a sub-formula \((\forall x \, \alpha)\) or \((\exists x \, \alpha)\) is the formula \(\alpha\).

An occurrence of a variable in a formula is *bound* if it lies in the scope of some quantifier of the same variable; otherwise it is *free*. In other words, a quantifier *binds* its variable within its scope.

*Example.* In formula \((\forall x \, (\exists y \, (x + y = z)))\), \(x\) is bound (by \(\forall x\)), \(y\) is bound (by \(\exists y\)), and \(z\) is free.

*Example.* In formula \((P(x) \land (\forall x \, (\neg Q(x))))\), the first occurrence of \(x\) is free and the last occurrence of \(x\) is bound.

(The variable symbol immediately after \(\exists\) or \(\forall\) is neither free nor bound.)
Free and Bound Variables

Formally, a variable occurs free in a formula $\alpha$ if and only if it is a member of the set $\text{FV}(\alpha)$ defined as follows.

1. If $\alpha$ is $P(t_1, \ldots, t_k)$, then $\text{FV}(\alpha) = \{ x \mid x \text{ appears in some } t_i \}$.
2. If $\alpha$ is $(\neg \beta)$, then $\text{FV}(\alpha) = \text{FV}(\beta)$.
3. If $\alpha$ is $(\beta \ast \gamma)$, then $\text{FV}(\alpha) = \text{FV}(\beta) \cup \text{FV}(\gamma)$.
4. If $\alpha$ is $(Qx \beta)$ (for $Q \in \{\forall, \exists\}$), then $\text{FV}(\alpha) = \text{FV}(\beta) - \{x\}$.

A formula has the same free variables as its parts, except that a quantified variable becomes bound.

A formula with no free variables is called a \textit{closed formula}, or a \textit{sentence}. 

Substitution

The notation $\alpha[t/x]$, for a variable $x$, a term $t$, and a formula $\alpha$, denotes the formula obtained from $\alpha$ by replacing each free occurrence of $x$ with $t$. Intuitively, it is the formula that answers the question, “What happens to $\alpha$ if $x$ has the value specified by term $t$?”

**Examples.**

- If $\alpha$ is the formula $E(f(x))$, then $\alpha[(y + y)/x]$ is $E(f(y + y))$.
- $\alpha[f(x)/x]$ is $E(f(f(x)))$.
- $E(f(x + y))[y/x]$ is $E(f(y + y))$.

Substitution does NOT affect bound occurrences of the variable.

- If $\beta$ is $(\forall x (E(f(x)) \land S(x, y)))$, then $\beta[g(x, y)/x]$ is $\beta$, because $\beta$ has no free occurrence of $x$. 
Example. Let $\beta$ be $(P(x) \land (\exists x \ Q(x)))$. What is $\beta[y/x]$?
Example. Let $\beta$ be $(P(x) \land (\exists x \ Q(x)))$. What is $\beta[y/x]$?

$\beta[y/x]$ is $(P(y) \land (\exists x \ Q(x)))$. Only the free $x$ gets substituted.
Examples: Substitution

Example. Let $\beta$ be $\left( P(x) \land (\exists x \ Q(x)) \right)$. What is $\beta[y/x]$?

$\beta[y/x]$ is $\left( P(y) \land (\exists x \ Q(x)) \right)$. Only the free $x$ gets substituted.

Example. What about $\beta[(y - 1)/z]$, where $\beta$ is $\left( \forall x \left( \exists y \ ((x + y) = z) \right) \right)$?
Examples: Substitution

Example. Let $\beta$ be $(P(x) \land (\exists x \ Q(x)))$. What is $\beta[y/x]$?

$\beta[y/x]$ is $(P(y) \land (\exists x \ Q(x)))$. Only the free $x$ gets substituted.

Example. What about $\beta[(y − 1)/z]$, where $\beta$ is 
$(\forall x (\exists y ((x + y) = z)))$?

At first thought, we might say $(\forall x (\exists y ((x + y) = (y − 1))))$. But there’s a problem—the free variable $y$ in the term $(y − 1)$ got “captured” by the quantifier $\exists y$.

We want to avoid this capture.
Avoiding Capture

Example. Formula $\alpha = S(x) \land \forall y (P(x) \rightarrow Q(y))$; term $t = f(y, y)$.

The leftmost $x$ can be substituted by $t$ since it is not in the scope of any quantifier, but substituting in $P(x)$ puts the variable $y$ into the scope of $\forall y$.

We can prevent capture of variables in two ways.

- Declare that a substitution is undefined in cases where capture would occur.
  One can often evade problems by a different choice of variable.
  (Above, we might be able to substitute $f(z, z)$ instead of $f(y, y)$. Or alter $\alpha$ to quantify some other variable.)

- Write the definition of substitution carefully, to prevent capture.

Huth and Ryan opt for the first method. We shall use the second.
Let \( x \) be a variable and \( t \) a term.

For a term \( u \), the term \( u[t/x] \) is \( u \) with each occurrence of the variable \( x \) replaced by the term \( t \).

For a formula \( \alpha \),

1. If \( \alpha \) is \( P(t_1, \ldots, t_k) \), then \( \alpha[t/x] \) is \( P(t_1[t/x], \ldots, t_k[t/x]) \).
2. If \( \alpha \) is \( \neg \beta \), then \( \alpha[t/x] \) is \( \neg \beta[t/x] \).
3. If \( \alpha \) is \( \beta \star \gamma \), then \( \alpha[t/x] \) is \( \beta[t/x] \star \gamma[t/x] \).
4. ...
For variable $x$, term $t$ and formula $\alpha$:

4. If $\alpha$ is $(Qx \, \beta)$, then $\alpha[t/x]$ is $\alpha$.

5. If $\alpha$ is $(Qy \, \beta)$ for some other variable $y$, then
   (a) If $y$ does not occur in $t$, then $\alpha[t/x]$ is $(Qy \, \beta[t/x])$.
   (b) Otherwise, select a variable $z$ that occurs in neither $\alpha$ nor $t$;
       then $\alpha[t/x]$ is $(Qz \, (\beta[z/y])[t/x])$.

The last case prevents capture by renaming the quantified variable to something harmless.

(Huth and Ryan specify that the substitution is undefined if capture would occur—case 5(b) above. With this more complex definition, one never has to add a condition regarding undefined substitutions. Substitution always behaves “the way it should”.)
Example, Revisited

Example. If $\alpha$ is $\left(\forall x \left(\exists y \left( x + y = z \right) \right) \right)$, what is $\alpha[(y - 1)/z]$?

This falls under case 5(b): the term to be substituted, namely $y - 1$, contains a variable $y$ quantified in formula $\alpha$.

Let $\beta$ be $(x + y = z)$; thus $\alpha$ is $\left(\forall x \left(\exists y \beta\right) \right)$.
Example. If $\alpha$ is \((\forall x (\exists y (x + y = z)))\), what is $\alpha[(y - 1)/z]$?

This falls under case 5(b): the term to be substituted, namely $y - 1$, contains a variable $y$ quantified in formula $\alpha$.

Let $\beta$ be $(x + y = z)$; thus $\alpha$ is $(\forall x (\exists y \beta))$.

Select a new variable, say $w$. Then

\[
\beta[w/y] \quad \text{is} \quad x + w = z,
\]

and

\[
\beta[w/y][(y - 1)/z] \quad \text{is} \quad (x + w) = (y - 1) .
\]

Thus the required formula $\alpha[(y - 1)/z]$ is

\[
(\forall x \exists w((x + w) = (y - 1))) .
\]
Semantics of Predicate Logic
Definition: Fix a set $L$ of constant symbols, function symbols, and predicate symbols. (The “language” of our formulas.)

An interpretation $I$ (for the set $L$) consists of

- A non-empty set $\text{dom}(I)$, called the domain (or universe) of $I$.
- For each constant symbol $c$, a member $c^I$ of $\text{dom}(I)$.
- For each function symbol $f^{(i)}$, an $i$-ary function $f^I$.
- For each predicate symbol $P^{(i)}$, an $i$-ary predicate $P^I$.

Huth and Ryan use the term “model” instead of “interpretation.” (Not a standard usage.)
For terms and formulas that contain no variables or quantifiers, an interpretation suffices to specify their meaning. The meaning arises in the obvious(?) fashion from the syntax of the term or formula.

**Definition:** Fix an interpretation $I$. For each term $t$ containing no variables, the value of $t$ under interpretation $I$, denoted $t^I$, is as follows.

- If $t$ is a constant $c$, the value $t^I$ is $c^I$.
- If $t$ is $f(t_1, \ldots, t_n)$, the value $t^I$ is $f^I(t_1^I, \ldots, t_n^I)$.

The value of a term is always a member of the domain of $I$. 
Formulas with Variable-Free Terms

Formulas get values in much the same fashion as terms, except that values of formulas lie in \{ \text{F, T} \}.

**Definition:** Fix an interpretation \( I \). For each formula \( \alpha \) containing no variables, the value of \( \alpha \) under interpretation \( I \), denoted \( \alpha^I \), is as follows.

- If \( \alpha \) is \( P(t_1, \ldots, t_n) \), then
  \[
  \alpha^I = \begin{cases} 
    \text{T} & \text{if } \langle t_1^I, \ldots, t_n^I \rangle \in P^I \\ 
    \text{F} & \text{otherwise.}
  \end{cases}
  \]

- If \( \alpha \) is \( (\neg \beta) \) or \( (\beta \star \gamma) \), then \( \alpha^I \) is determined by \( \beta^I \) and \( \gamma^I \) in the same way as for propositional logic.
Examples

Let 0 be a constant symbol, \( f^{(1)} \) a function symbol and \( E^{(1)} \) a predicate symbol. Thus \( E(f(0)) \) and \( E(f(f(0))) \) are both formulas.

Consider an interpretation \( I \) with

- **Domain:** \( \mathbb{N} \), the natural numbers
- \( 0^I \): zero
- \( f^I \): successor; \( \{ \langle x, x + 1 \rangle \mid x \in \mathbb{N} \} \)
- \( E^I \): “is even”; \( \{ 2y \mid y \in \mathbb{N} \} \)

Terms get numerical values: \( f(0)^I \) is 1 and \( f(f(0))^I \) is 2.

Formula \( E(f(0)) \) means “1 is even”, and \( E(f(0))^I = F \).

Formula \( E(f(f(0))) \) means “2 is even”, and \( E(f(f(0)))^I = T \).

What about some other interpretation?
Let $J$ be the interpretation with

- **Domain:** $\mathbb{Q}$, the rational numbers
- $0^J$: two
- $f^J$: halving; $\{ \langle x, x/2 \rangle \mid x \in \mathbb{Q} \}$
- $E^J$: “is an integer”; $\{ x \mid x \in \mathbb{Z} \}$

$E(f(0))$ means “1 is an integer”, and $E(f(0))^J$ is $\mathbf{T}$.

$E(f(f(0)))$ means “1/2 is an integer”, and $E(f(f(0)))^J$ is $\mathbf{F}$.

**Exercise:** in both $I$ and $J$, the formula $\left( E(f(f(0))) \land E(f(0)) \right)$ receives value $\mathbf{F}$. Find another interpretation which gives it the value $\mathbf{T}$. 
“Gotchas”

Two often-overlooked points about interpretations.

1. There is NO intrinsic meaning for predicate, function or constant symbols.

   “1 + 2 = 3” might mean that one plus two equals three—but only if we specify that interpretation. Any interpretation of constants 1, 2, and 3, function symbol $+(2)$ and predicate symbol $=(2)$ is possible.

2. Functions must be defined at every point in the domain. (I.e., they must be $total$.)

   If we have language with a binary function symbol “$−$”, we cannot specify an interpretation with domain $\mathbb{N}$ and subtraction for “$−$”. Subtraction is not total on $\mathbb{N}$. 
Predicate Logic Adds to Propositional Logic

In propositional logic, semantics was described in terms of valuations to propositional atoms.

Predicate Logic includes more ingredients (i.e., predicates, functions, variables, terms, constants, etc.) and, hence, the semantics for Predicate Logic must account for all of the ingredients.

We already saw the concept of an interpretation, which specifies the domain and the identities of the constants, predicates and functions.

Formulas that include variables, and perhaps quantifiers, require additional information, known as an environment (or assignment).
A Predicate *environment* is a function that assigns a value in the domain to each variable.

**Example.** With the domain $\mathbb{N}$, we might have environment $E_1$ given by $E_1(x) = 9$ and $E_1(y) = 2$.

If the interpretation specifies $<$ is less-than, then $x < y$ gets value false.

**Example.** With the domain of fictional animals, we might have $E_2(x) = Tweety$ and $E_2(y) = Nemo$.

If the interpretation specifies $<$ is “was created before”, then $x < y$ gets value true.
**Example:** Let $\alpha_1$ be $P(c)$ (where $c$ is a constant), and let $\alpha_2$ be $P(x)$ (where $x$ a variable).

Let $I$ be the interpretation with domain $\mathbb{N}$, $c^I = 2$ and $P^I = \text{“is even”}$. Then $\alpha_1^I = T$, but $\alpha_2^I$ is undefined.

To give $\alpha_2$ a value, we must also specify an environment. For example, if $E(x) = 2$, then $\alpha_2^{(I,E)} = T$.

If we wish, we can consider a formula such as $\alpha_2$ that contains a free variable $x$ as expressing a function: the function that maps $E(x)$ to $\alpha_2^{(I,E)}$. 
Meaning of Terms

The combination of an interpretation and an environment supplies a value for every term.

**Definition:** Fix an interpretation $I$ and environment $E$. For each term $t$, the value of $t$ under $I$ and $E$, denoted $t^{(I,E)}$, is as follows.

- If $t$ is a constant $c$, the value $t^{(I,E)}$ is $c^I$.
- If $t$ is a variable $x$, the value $t^{(I,E)}$ is $x^E$.
- If $t$ is $f(t_1, \ldots, t_n)$, the value $t^{(I,E)}$ is $f^I(t_1^{(I,E)}, \ldots, t_n^{(I,E)})$.

To extend this definition to formulas, we must consider quantifiers.

But first, a few examples.
Meaning of Terms—Example

Example. Suppose a language has constant symbol 0, a unary function \( s \), and a binary function \( + \). We shall write \( + \) in infix position: \( x + y \) instead of \( +(x, y) \).

The expressions \( s(s(0) + s(x)) \) and \( s(x + s(x + s(0))) \) are both terms.

The following are examples of interpretations and environments.

- \( dom\{I\} = \{0, 1, 2, \ldots\} \), \( 0^I = 0 \), \( s^I \) is the successor function and \( +^I \) is the addition operation. Then, if \( E(x) = 3 \), the terms get values
  \[
  (s(s(0) + s(x)))^{(I,E)} = 6 \quad \text{and} \quad (s(x + s(x + s(0))))^{(I,E)} = 9.
  \]
• $\text{dom}\{J\}$ is the collection of all words over the alphabet \(\{a, b\}\),
\(0^J = a\),
\(s^J\) appends \(a\) to the end of a string, and
\(+^J\) is concatenation.

Let \(E(x) = aba\). Then

\[
\left(s(s(0) + s(x))\right)^{(J,E)} = aaabaaa
\]

and

\[
\left(s(x + s(x + s(0)))\right)^{(J,E)} = abaabaaaaa .
\]
Quantified Formulas

To evaluate the truthfulness of a formula $(\forall x \alpha)$ (resp. $(\exists x \alpha)$), we should check whether $\alpha$ holds for every (resp., for some) value $a$ in the domain.

How can we express this precisely?

**Definition:** For any environment $E$ and domain element $d$, the environment “$E$ with $x$ re-assigned to $d$”, denoted $E[x \mapsto d]$, is given by

$$E[x \mapsto d](y) = \begin{cases} 
  d & \text{if } y \text{ is } x \\
  E(y) & \text{if } y \text{ is not } x.
\end{cases}$$
Values of Quantified Formulas

**Definition:** The values of $(\forall x \, \alpha)$ and $(\exists x \, \alpha)$ are given by

- $(\forall x \, \alpha)(I,E) = \begin{cases} 
  T & \text{if } \alpha(I,E[x \mapsto d]) = T \text{ for every } d \in \text{dom}(I) \\
  F & \text{otherwise} 
\end{cases}$

- $(\exists x \, \alpha)(I,E) = \begin{cases} 
  T & \text{if } \alpha(I,E[x \mapsto d]) = T \text{ for some } d \in \text{dom}(I) \\
  F & \text{otherwise} 
\end{cases}$

Note: The values of $(\forall x \, \alpha)(I,E)$ and $(\exists x \, \alpha)(I,E)$ do not depend on the value of $E(x)$.

The value $E(x)$ only matters for free occurrences of $x$. 
Example. Let $\text{dom}(I) = \{a, b\}$ and $P^I = \{\langle a, a \rangle, \langle a, b \rangle, \langle b, b \rangle\}$.

Let $E(x) = a$ and $E(y) = b$. We have

- $P(x, x)^{(I,E)} = \top$, since $\langle E(x), E(x) \rangle = \langle a, a \rangle \in P^I$.
- $P(y, x)^{(I,E)} = \bot$, since $\langle E(y), E(x) \rangle = \langle b, a \rangle \notin P^I$.
- $(\exists y P(y, x))^{(I,E)} = \top$, since $P(y, x)^{(I,E[y \mapsto a])} = \top$.
  
  (That is, $\langle E[y \mapsto a](y), E[y \mapsto a](x) \rangle = \langle a, a \rangle \in P^I$).
- What is $(\forall x (\forall y P(x, y)))^{(I,E)}$?
Example. Let \( \text{dom}(I) = \{a, b\} \) and \( P^I = \{\langle a, a\rangle, \langle a, b\rangle, \langle b, b\rangle\} \).

Let \( E(x) = a \) and \( E(y) = b \).

• What is \((\forall x (\forall y P(x, y)))^{(I,E)}\)?

Since \( \langle b, a \rangle \notin P^I \), we have

\[
P(x, y)^{(I,E[x\mapsto b][y\mapsto a])} = F ,
\]

and thus

\[
(\forall x (\forall y P(x, y)))^{(I,E)} = F .
\]
Example. Let \( \text{dom}(I) = \{a, b\} \) and \( P^I = \{\langle a, a \rangle, \langle a, b \rangle, \langle b, b \rangle\} \).

Let \( E(x) = a \) and \( E(y) = b \).

- What is \( (\forall x (\forall y P(x, y)))^{(I,E)} \)?

Since \( \langle b, a \rangle \notin P^I \), we have

\[
P(x, y)^{(I,E[x \mapsto b][y \mapsto a])} = F
\]

and thus

\[
(\forall x (\forall y P(x, y)))^{(I,E)} = F
\]

- What about \( (\forall x (\exists y P(x, y)))^{(I,E)} \)?
In the previous example, we wrote

\[ P(x, y)^{(I, E[x\mapsto b][y\mapsto a])} = F. \]

Why did we not write simply

\[ P(b, a) = F \]

or perhaps

\[ P(b, a)^{(I, E)} = F? \]
In the previous example, we wrote

\[ P(x, y)(I, E[x \mapsto b][y \mapsto a]) = F. \]

Why did we not write simply

\[ P(b, a) = F \]

or perhaps

\[ P(b, a)(I, E) = F? \]

Because “\(P(b, a)\)” is not a formula. The elements a and b of \(\text{dom}(I)\) are not symbols in the language; they cannot appear in a formula.
Satisfaction of Formulas

An interpretation $I$ and environment $E$ satisfy a formula $\alpha$, denoted $I \models_E \alpha$, if $\alpha^{(I,E)} = T$;
they do not satisfy $\alpha$, denoted $I \not\models_E \alpha$, if $\alpha^{(I,E)} = F$.

<table>
<thead>
<tr>
<th>Form of $\alpha$</th>
<th>Condition for $I \models_E \alpha$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P(t_1, \ldots, t_k)$</td>
<td>$\langle t_1^{(I,E)}, \ldots, t_k^{(I,E)} \rangle \in PI$</td>
</tr>
<tr>
<td>$(\neg \beta)$</td>
<td>$I \not\models_E \beta$</td>
</tr>
<tr>
<td>$(\beta \land \gamma)$</td>
<td>both $I \models_E \beta$ and $I \models_E \gamma$</td>
</tr>
<tr>
<td>$(\beta \lor \gamma)$</td>
<td>either $I \models_E \beta$ or $I \models_E \gamma$ (or both)</td>
</tr>
<tr>
<td>$(\beta \rightarrow \gamma)$</td>
<td>either $I \not\models_E \beta$ or $I \models_E \gamma$ (or both)</td>
</tr>
<tr>
<td>$(\forall x \beta)$</td>
<td>for every $a \in \text{dom}(I)$, $I \models_{E[x \mapsto a]} \beta$</td>
</tr>
<tr>
<td>$(\exists x \beta)$</td>
<td>there is some $a \in \text{dom}(I)$ such that $I \models_{E[x \mapsto a]} \beta$</td>
</tr>
</tbody>
</table>

If $I \models_E \alpha$ for every $E$, then $I$ satisfies $\alpha$, denoted $I \models \alpha$. 
Example. Consider the formula $(\exists y \, P(x, y \oplus y))$.

(For $P$ a binary predicate and $\oplus$ a binary function.)

Suppose $\text{dom}(I) = \{1, 2, 3, \ldots\}$,

$\oplus^I$ is the addition operation, and

$P^I$ is the equality predicate.

Then $I \models_E (\exists y \, P(x, y \oplus y))$ iff $E(x)$ is an even number.
Validity and Satisfiability

Validity and satisfiability of formulas have definitions analogous to the ones for propositional logic.

**Definition:** A formula $\alpha$ is

- **valid** if every interpretation and environment satisfy $\alpha$; that is, if $I \models_E \alpha$ for every $I$ and $E$,
- **satisfiable** if some interpretation and environment satisfy $\alpha$; that is, if $I \models_E \alpha$ for some $I$ and $E$, and
- **unsatisfiable** if no interpretation and environment satisfy $\alpha$; that is, if $I \not\models_E \alpha$ for every $I$ and $E$.

(The term “tautology” is not used in predicate logic.)
Example: Satisiability and Validity

Let $\alpha$ be the formula $P(f(g(x), g(y)), g(z))$. The formula is satisfiable:

- $dom(I)$: $\mathbb{N}$
- $f^I$: summation
- $g^I$: squaring
- $P^I$: equality
- $E(x) = 3$, $E(y) = 4$ and $E(z) = 5$.

$\alpha$ is not valid. (Why?)
The universal and existential quantifiers may be understood respectively as generalizations of conjunction and disjunction. If the domain $D = \{a_1, \ldots, a_k\}$ is finite then:

For all $x$, $P(x)$ iff $P(a_1)$ and ... and $P(a_k)$

There exists $x$, $P(x)$ iff $P(a_1)$ or ... or $P(a_k)$

where $P$ is a predicate (a property).
Relevance Lemma

Lemma:
Let $\alpha$ be a Predicate formula, let $I$ be an interpretation, and let $E_1$ and $E_2$ be two environments such that

$$E_1(x) = E_2(x) \text{ for every } x \text{ that occurs free in } \alpha.$$  

Then

$$I \models_{E_1} \alpha \text{ if and only if } I \models_{E_2} \alpha.$$  

Proof by induction on the structure of $\alpha$. 

Suppose $\Sigma$ is a set of formulas and $\alpha$ is a formula. We say that $\Sigma$ is a \textit{semantically entails} $\alpha$, written as $\Sigma \models \alpha$, iff for any interpretation $I$ and environment $E$, we have $I \models_{E} \Sigma$ implies $I \models_{E} \alpha$.

$\emptyset \models \alpha$ means that $\alpha$ is valid.
Example: Semantic Entailment

Example: Show that $\emptyset \models \left( (\forall x (\alpha \rightarrow \beta)) \rightarrow ((\forall x \alpha) \rightarrow (\forall x \beta)) \right)$.

Proof by contradiction. Suppose there are $I$ and $E$ such that

$$I \not\models_E \left( (\forall x (\alpha \rightarrow \beta)) \rightarrow ((\forall x \alpha) \rightarrow (\forall x \beta)) \right).$$

Then we must have $I \models_E (\forall x (\alpha \rightarrow \beta))$ and $I \not\models_E ((\forall x \alpha) \rightarrow (\forall x \beta))$; the second gives $I \models_E (\forall x \alpha)$ and $I \not\models_E (\forall x \beta)$.

Using the definition of $\models$ for formulas with $\forall$, we have for every $a \in \text{dom}(I)$, $I \models_{E[x \mapsto a]} (\alpha \rightarrow \beta)$ and $I \models_{E[x \mapsto a]} \alpha$.

Thus also $I \models_{E[x \mapsto a]} \beta$ for every $a \in \text{dom}(I)$.

Thus $I \models_E \forall x \beta$, a contradiction.
Example. Show that \((∀x(¬γ)) \models (¬(∃x γ))\).
Example II: Entailment

**Example.** Show that \((\forall x(\neg \gamma)) \models (\neg (\exists x \gamma))\).

Suppose that \(I \models_E (\forall x(\neg \gamma))\). By definition, this means for every \(a \in \text{dom}(I)\), \(I \models_{E[x \mapsto a]} (\neg \gamma)\).

Again by definition (for a formula with \(\neg\)), this is equivalent to for every \(a \in \text{dom}(I)\), \(I \not\models_{E[x \mapsto a]} \gamma\)

and there is no \(a \in \text{dom}(I)\) such that \(I \models_{E[x \mapsto a]} \gamma\).

Also, this last is the definition of \(I \models_E (\neg (\exists x \gamma))\), as required.
Example: Show that, in general,

\[ ((\forall x \alpha) \rightarrow (\forall x \beta)) \not\equiv (\forall x (\alpha \rightarrow \beta)) \]

(That is, find \( \alpha \) and \( \beta \) such that consequence does not hold.)
Example: Show that, in general,

\[ ((\forall x \alpha) \rightarrow (\forall x \beta)) \not\equiv (\forall x (\alpha \rightarrow \beta)) \]

(That is, find \( \alpha \) and \( \beta \) such that consequence does not hold.)

Key idea: \( \varphi_1 \rightarrow \varphi_2 \) yields true whenever \( \varphi_1 \) is false.

Let \( \alpha \) be \( P(x) \). Let \( I \) have domain \( \{a, b\} \) and \( P^I = \{a\} \). Then \( I \models (\forall x \alpha) \rightarrow (\forall x \beta) \) for any \( \beta \). (Why?)
Example: Show that, in general,

\[ ((\forall x \alpha) \rightarrow (\forall x \beta)) \nvDash (\forall x (\alpha \rightarrow \beta)) \]

(That is, find \( \alpha \) and \( \beta \) such that consequence does not hold.)

Key idea: \( \varphi_1 \rightarrow \varphi_2 \) yields true whenever \( \varphi_1 \) is false.

Let \( \alpha \) be \( P(x) \). Let \( I \) have domain \{a, b\} and \( P^I = \{a\} \). Then
\( I \vDash (\forall x \alpha) \rightarrow (\forall x \beta) \) for any \( \beta \). (Why?)

To obtain \( I \nvDash \forall x (\alpha \rightarrow \beta) \), we can use \( \neg P(x) \) for \( \beta \). (Why?)

Thus \( ((\forall x \alpha) \rightarrow (\forall x \beta)) \nvDash (\forall x (\alpha \rightarrow \beta)) \), as required. (Why?)
Example: for any formula $\alpha$ and term $t$,

$$\emptyset \models ((\forall x \alpha) \rightarrow \alpha[t/x]) .$$

Recall that functions must be total!
Proofs in Predicate Logic
Using Natural Deduction
Natural Deduction for Predicate Logic extends Natural Deduction for propositional logic by including rules for introduction and elimination of quantifiers.

Other proof techniques and tricks remain the same as natural deduction for propositional logic.
Elimination of ∀ and introduction of ∃ are fairly straightforward.

<table>
<thead>
<tr>
<th>Name</th>
<th>⊢-notation</th>
<th>inference notation</th>
</tr>
</thead>
<tbody>
<tr>
<td>∀-elimination (∀e)</td>
<td>If ( \Sigma \vdash \forall x \alpha ) then ( \Sigma \vdash \alpha[t/x] )</td>
<td>( \forall x \alpha ) [\alpha[t/x]]</td>
</tr>
<tr>
<td>∃-introduction (∃i)</td>
<td>If ( \Sigma \vdash \alpha[t/x] ), then ( \Sigma \vdash \exists x \alpha )</td>
<td>( \exists x \alpha ) [\alpha[t/x]]</td>
</tr>
</tbody>
</table>

Given that a formula is true for every value of \( x \), conclude it is true for any particular value, such as that of \( t \).

Given that a formula is true for a particular value (of \( t \)), conclude it is true for some value.
Example: $\forall e$

“All fish can swim. Nemo is a fish. Therefore, Nemo can swim.”

In Predicate Logic: show that
\[ \{ \forall x(F(x) \rightarrow S(x)), F(Nemo) \} \vdash S(Nemo) \]

Proof:

1. $\forall x(F(x) \rightarrow S(x))$  Premise
2. $F(Nemo)$  Premise
3. $F(Nemo) \rightarrow S(Nemo)$  $\forall e$: 1
4. $S(Nemo)$  $\rightarrow e$: 2, 3

The proof doesn’t care what $F$ and $S$ mean. Fishiness and swimming ability really have nothing to do with the argument.
Example: \( \exists i \)

**Example.** Show \( \neg P(y) \vdash \exists x \left( P(x) \rightarrow Q(y) \right) \).

1. \( \neg P(y) \)  
   **Premise**
2. \( P(y) \)  
   **Assumption**
3. \( \bot \)  
   \( \neg e: 2, 1 \)
4. \( Q(y) \)  
   \( \bot e: 3 \)
5. \( P(y) \rightarrow Q(y) \)  
   \( \rightarrow i: 2-4 \)
6. \( \exists x \left( P(x) \rightarrow Q(y) \right) \)  
   \( \exists i: 5 \)
Note to the example

The general form of rule $\exists i$:

\[
\frac{\alpha[t/x]}{\exists x \alpha}
\]

Use in the previous example:

\[
\frac{P(y) \rightarrow Q(y)}{\exists x (P(x) \rightarrow Q(y))}
\]

We took $P(x) \rightarrow Q(y)$ for $\alpha$.

However, knowing what $\alpha[t/x]$ is, does not determine what $\alpha$ is.

We could also take $P(x) \rightarrow Q(x)$ for $\alpha$; thus the derivation step would be

\[
\frac{P(y) \rightarrow Q(y)}{\exists x (P(x) \rightarrow Q(x))}.
\]

But the formula $\exists x (P(x) \rightarrow Q(x))$ is not what we wanted to prove.
The $\forall$-introduction rule follows ordinary mathematical usage. To prove a property holds for all integers, one often starts with

Let $x$ be an integer.

This means the same as

Assume that the variable “$x$” refers to an integer.

Then one proves that $x$ has the property.

Since we know nothing about the value $x$, except that it is an integer, this justifies that every integer has the property.

One could also start the proof with

Let $x$ be anything. If $x$ is an integer, then...

The conclusion is essentially the same.
Definition: a variable is fresh in a subproof if it occurs nowhere outside the box of the subproof.

Freshness captures the notion of “know nothing about”.

<table>
<thead>
<tr>
<th>Name</th>
<th>⊢-notation</th>
<th>inference notation</th>
</tr>
</thead>
</table>
| ∀-introduction  | If \( \Sigma \vdash \alpha[u/x] \) and \( u \) not free in \( \Sigma \) or \( \alpha \), then \( \Sigma \vdash \forall x \alpha \) | \[
\begin{array}{c}
\alpha[u/x] \\
\hline
u \text{ fresh}
\end{array}
\]
|                 |            | \( \forall x \alpha \) |

In words: in order to prove \( \forall x \alpha(x) \), prove \( \alpha(u) \) for arbitrary \( u \).
Example: Use of \( \forall \)

**Example.** Show that \( \neg \exists x \alpha \vdash \forall x \neg \alpha \), for any \( \alpha \).

1. \( \neg \exists x \alpha \) \hspace{1cm} \text{Premise}
2. \( u \) fresh

\[
\neg \alpha[u/x] \quad ??
\]

\( \forall x \neg \alpha \) \hspace{1cm} \forall i: 2–6

Note: “\( u \) fresh” means we choose any variable not in \( \alpha \).
Example: Use of $\forall i$

**Example.** Show that $\neg \exists x \alpha \vdash \forall x \neg \alpha$, for any $\alpha$.

1. $\neg \exists x \alpha$  
   Premise

2. $u$ fresh

3. $\alpha[u/x]$  
   Assumption

4. $\exists x \alpha$  
   $\exists i$: 3

5. $\bot$  
   $\neg e$: 1, 4

6. $\neg \alpha[u/x]$  
   $\neg i$: 3–5

7. $\forall x \neg \alpha$  
   $\forall i$: 2–6

Note: “$u$ fresh” means we choose any variable not in $\alpha$. 
Example: Another use of $\forall i$

Show that $\forall x (\alpha \rightarrow \beta) \vdash (\forall x \alpha) \rightarrow (\forall x \beta)$.

1. $\forall x (\alpha \rightarrow \beta)$  Premise

$$(\forall x \alpha) \rightarrow (\forall x \beta) \rightarrow i??$$

Note: do not apply rule $\forall e$ until you know which term to use.
Example: Another use of $\forall i$

Show that $\forall x (\alpha \rightarrow \beta) \vdash (\forall x \alpha) \rightarrow (\forall x \beta)$.

1. $\forall x (\alpha \rightarrow \beta)$  Premise

2. $\forall x \alpha$  Assumption

3. $u$ fresh

4. $\alpha[u/x] \rightarrow \beta[u/x]$ $\forall e: 1$

5. $\alpha[u/x]$ $\forall e: 2$

6. $\beta[u/x]$ ??

7. $\forall x \beta$ $\forall i$??

8. $(\forall x \alpha) \rightarrow (\forall x \beta)$ $\rightarrow i$??

Note: do not apply rule $\forall e$ until you know which term to use.
Example: Another use of $\forall i$

Show that $\forall x (\alpha \rightarrow \beta) \vdash (\forall x \alpha) \rightarrow (\forall x \beta)$.

1. $\forall x (\alpha \rightarrow \beta)$ Premise
2. $\forall x \alpha$ Assumption
3. $u$ fresh
4. $\alpha[u/x] \rightarrow \beta[u/x]$ $\forall$-Introduction $e: 1$
5. $\alpha[u/x]$ $\forall$-Introduction $e: 2$
6. $\beta[u/x]$ ??
7. $\forall x \beta$ $\forall$-Introduction ??
8. $(\forall x \alpha) \rightarrow (\forall x \beta)$ $\rightarrow$ $i$ ??

Note: do not apply rule $\forall e$ until you know which term to use.
Example: Another use of $\forall i$

Show that $\forall x (\alpha \rightarrow \beta) \vdash (\forall x \alpha) \rightarrow (\forall x \beta)$.

1. $\forall x (\alpha \rightarrow \beta)$ Premise
2. $\forall x \alpha$ Assumption
3. $u$ fresh
4. $\alpha[u/x] \rightarrow \beta[u/x]$ $\forall e$: 1
5. $\alpha[u/x]$ $\forall e$: 2
6. $\beta[u/x]$ $\rightarrow e$: 4, 5
7. $\forall x \beta$ $\forall i$: 3–6
8. $(\forall x \alpha) \rightarrow (\forall x \beta)$ $\rightarrow i$: 2–7

Note: do not apply rule $\forall e$ until you know which term to use.
### Elimination of an Existential Quantifier

<table>
<thead>
<tr>
<th>Name</th>
<th>(\vdash)-notation</th>
<th>inference notation</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\exists)-elimination ((\exists)e)</td>
<td>If (\Sigma, \alpha[u/x] \vdash \beta), with (u) fresh, then (\Sigma, \exists x \alpha \vdash \beta)</td>
<td>(\alpha[u/x], u ) fresh (\exists x \alpha) (\vdash \beta)</td>
</tr>
</tbody>
</table>
Example: Use of \( \exists e \)

Example. Show that \( \exists x \ R(x) \vdash \exists y \ R(y) \).

1. \( \exists x \ R(x) \)  
   \hspace{1cm} \text{Premise}

2. \( R(u), u \text{ fresh} \)  
   \hspace{1cm} \text{Assumption}

3. \( \exists y \ R(y) \)  
   \hspace{1cm} \exists i: 2 \ (\text{term } u)

4. \( \exists y \ R(y) \)  
   \hspace{1cm} \exists e: 1, 2–3
Extending the example?

Clearly, the previous proof did not depend on the particular relation $R$ that we used. Can we do the same proof for arbitrary formulas?

Does $\exists x \alpha \vdash \exists y \alpha[y/x]$ hold?

1. $\exists x \alpha$                  Premise
2. $\alpha[u/x], u$ fresh Assumption
3. $\alpha[y/x][u/y]$        ???
4. $\exists y \alpha[y/x]$       $\exists i$: 3 (term $u$)
5. $\exists y \alpha[y/x]$       $\exists e$: 1, 2–4

Is the formula on line 2 the same as the one on line 3?
Extending the example?

Clearly, the previous proof did not depend on the particular relation $R$ that we used. Can we do the same proof for arbitrary formulas?

Does $\exists x \alpha \vdash \exists y \alpha[y/x]$ hold?

1. $\exists x \alpha$  Premise
2. $\alpha[u/x], u$ fresh Assumption
3. $\alpha[y/x][u/y]$  ????
4. $\exists y \alpha[y/x]$  $\exists i: 3$ (term $u$)
5. $\exists y \alpha[y/x]$  $\exists e: 1, 2–4$

Is the formula on line 2 the same as the one on line 3?

If $y$ is free in $\alpha$, then no — the derivation fails. But otherwise, it works.
Example: \( \exists \) and \( \forall \) together

**Example.** Show that \( \exists x \neg \alpha \vdash \neg \forall x \alpha \).

1. \( \exists x \neg \alpha \)                 Premise

\[ \neg \forall x \alpha \] \( \exists \text{-} \text{Elimination} \) ??
Example: $\exists$ and $\forall$ together

Example. Show that $\exists x \neg \alpha \vdash \neg \forall x \alpha$.

1. $\exists x \neg \alpha$  Premise
2. $\neg \alpha[u/x], u$ fresh Assumption

\[ \neg \forall x \alpha \quad \neg \text{i ??} \]

7. $\neg \forall x \alpha$  $\exists$e ??
Example: \( \exists \) and \( \forall \) together

Example. Show that \( \exists x \neg \alpha \vdash \neg \forall x \alpha \).

1. \( \exists x \neg \alpha \) Premise
2. \( \neg \alpha[u/x], u \text{ fresh} \) Assumption
3. \( \forall x \alpha \) Assumption
4. \( \alpha[u/x] \) \( \forall e: 3 \)
5. \( \bot \) \( \neg e: 4, 2 \)
6. \( \neg \forall x \alpha \) \( \neg i: 3–5 \)
7. \( \neg \forall x \alpha \) \( \exists e: 1, 2–6 \)

Natural Deduction for Predicate Logic
\( \exists \)-Elimination
Example: \( \forall e \) and \( \exists i \) together, again

We can interchange the quantifiers in the previous deduction.

**Example.** Show \( \forall x \neg \alpha \vdash \neg \exists x \alpha \).

<table>
<thead>
<tr>
<th>Step</th>
<th>Formula</th>
<th>Justification</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>( \forall x \neg \alpha )</td>
<td>Premise</td>
</tr>
<tr>
<td>2</td>
<td>( \exists x \alpha )</td>
<td>Assumption</td>
</tr>
<tr>
<td>3</td>
<td>( \alpha[u/x] ) (( u ) fresh)</td>
<td>Assumption</td>
</tr>
<tr>
<td>4</td>
<td>( \neg \alpha[u/x] )</td>
<td>( \forall e: 1 )</td>
</tr>
<tr>
<td>5</td>
<td>( \bot )</td>
<td>( \neg e: 3, 4 )</td>
</tr>
<tr>
<td>6</td>
<td>( \bot )</td>
<td>( \exists e: 2, 3–5 )</td>
</tr>
<tr>
<td>7</td>
<td>( \neg \exists x \alpha )</td>
<td>( \neg i: 2–6 )</td>
</tr>
</tbody>
</table>
Quantifiers and Negation: The final case

So far, we have shown \( \neg \exists x \alpha \vdash \forall x \neg \alpha \), \( \forall x \neg \alpha \vdash \neg \exists x \alpha \), and \( \exists x \neg \alpha \vdash \neg \forall x \alpha \).

**Example.** Show that \( \neg \forall x \alpha \vdash \exists x \neg \alpha \).

1. \( \neg \forall x \alpha \) Premise

   \( \neg \alpha[t/x] \) ??

   \( \exists x \neg \alpha \) \( \exists i: ?? \)
Quantifiers and Negation: The final case

So far, we have shown \( \neg \exists x \alpha \vdash \forall x \neg \alpha \),
\( \forall x \neg \alpha \vdash \neg \exists x \alpha \), and
\( \exists x \neg \alpha \vdash \neg \forall x \alpha \).

**Example.** Show that \( \neg \forall x \alpha \vdash \exists x \neg \alpha \).

1. \( \neg \forall x \alpha \)  Premise

\[ \neg \alpha[t/x] \] ??

\( \exists x \neg \alpha \)  \( \exists i: \) ??

For what term \( t \) can we prove \( \neg \alpha[t/x] \)?
Quantifiers and Negation: The final case

So far, we have shown $\neg \exists x \alpha \vdash \forall x \neg \alpha$, $\forall x \neg \alpha \vdash \neg \exists x \alpha$, and $\exists x \neg \alpha \vdash \neg \forall x \alpha$.

**Example.** Show that $\neg \forall x \alpha \vdash \exists x \neg \alpha$.

1. $\neg \forall x \alpha$ Premise

   $\neg \alpha[t/x]$ ??

   $\exists x \neg \alpha$ $\exists i$: ??

For what term $t$ can we prove $\neg \alpha[t/x]$?

There is no such $t$!

We need to try something cleverer....
The Final Case: A full proof

Example. Show that \( \neg \forall x \alpha \vdash \exists x \neg \alpha \).

<p>| | | | | | | | | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>((\neg (\forall x \alpha)))</td>
<td>Premise</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.</td>
<td>((\neg (\exists x (\neg \alpha))))</td>
<td>Assumption</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.</td>
<td>(u) fresh</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4.</td>
<td>((\neg \alpha[u/x]))</td>
<td>Assumption</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.</td>
<td>((\exists x (\neg \alpha)))</td>
<td>(\exists)i: 4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6.</td>
<td>(\bot)</td>
<td>(\neg)e: 5, 2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7.</td>
<td>((\neg (\neg \alpha[u/x])))</td>
<td>(\neg)i: 4–6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8.</td>
<td>(\alpha[u/x])</td>
<td>(\neg\neg)e: 7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9.</td>
<td>((\forall x \alpha))</td>
<td>(\forall)i: 3–8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10.</td>
<td>(\bot)</td>
<td>(\neg)e: 9, 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11.</td>
<td>((\neg (\neg (\exists x (\neg \alpha)))))</td>
<td>(\neg)i: 2–10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12.</td>
<td>((\exists x (\neg \alpha)))</td>
<td>(\neg\neg)e: 11</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Repeated Quantifiers

The rules for elimination and introduction of quantifiers can be generalized to multiple quantifiers.

Let $x_1, \ldots, x_n$ be $n$ distinct variables.

- If $\Sigma \vdash \forall x_1 \cdots \forall x_n \alpha$, then $\Sigma \vdash \alpha[t_1/x_1] \cdots [t_n/x_n]$.
- If $\Sigma \vdash \alpha[t_1/x_1] \cdots [t_n/x_n]$, for terms $t_1, \ldots, t_n$, then $\Sigma \vdash \exists x_1 \cdots \exists x_n \alpha$.
- If $\Sigma \vdash \alpha[u_1/x_1] \cdots [u_n/x_n]$, with variables $u_1, \ldots, u_n$ fresh, then $\Sigma \vdash \forall x_1 \cdots \forall x_n \alpha$.
- If $\Sigma \vdash \exists x_1 \cdots \exists x_n \alpha$ and $\Sigma \cup \{\alpha[u_1/x_1] \cdots [u_n/x_n] \vdash \beta$, with $u_1, \ldots, u_n$ fresh, then $\Sigma \vdash \beta$. 

Natural Deduction for Predicate Logic

$\exists$-Elimination
Example. Show that \( \forall x \forall y A(x, y) \vdash \forall y \forall x A(x, y) \).

1. \( \forall x \forall y A(x, y) \)  
   Premise
2. \( u, v \) fresh
3. \( A(u, v) \)  
   \( \forall e (\times 2): 1 \)
4. \( \forall y \forall x A(x, y) \)  
   \( \forall i (\times 2): 3 \)
Exercise. Show that

\[ \{ \forall x (Q(x) \rightarrow R(x)), \exists x (P(x) \land Q(x)) \} \vdash \exists x (P(x) \land R(x)) . \]

Left to you.
Soundness and Completeness of Natural Deduction

**Theorem.**

- Natural Deduction is sound for Predicate Logic: if $\Sigma \vdash \alpha$, then $\Sigma \models \alpha$.
- Natural Deduction is complete for Predicate Logic: if $\Sigma \models \alpha$, then $\Sigma \vdash \alpha$.

Proof outline:

**Soundness:** Each application of a rule is sound. By induction, any finite number of rule applications is sound.

**Completeness:** We shall show the contrapositive:

$$\text{if } \Sigma \nvdash \alpha, \text{ then } \Sigma \nmodels \alpha.$$  

We shall not give the full proof, but we will sketch the main points.
Soundness of $\forall$-Elimination and $\exists$-Introduction

**Claim:** For any formula $\varphi$, variable $x$ and term $t$,

$$\forall x \varphi \models \varphi[t/x] \quad \text{and} \quad \varphi[t/x] \models \exists x \varphi.$$ 

**Proof:** Suppose $I \models_E (Qx \varphi)$; i.e., for (every/some) $d \in \text{dom}(I)$,

$$\varphi(I, E[x \mapsto d]) = T.$$ 

Since $t^{(I, E)}$ is value in the domain, it suffices to show

**Claim II:** For every formula $\varphi$, variable $x$ and term $t$,

$$\varphi[t/x]^{(I, E)} = \varphi(I, E[x \mapsto t^{(I, E)}]).$$

To prove this second claim, use the definition of substitution.

(Left to you. Cases 4, 5(a) and 5(b) of the definition make it work.)
Some Details from the Proof of Claim II

One of the cases: \( \varphi \) is a quantified formula, say \( \exists y \alpha \).

We have \( (\exists y \alpha)^{(I,E[x \mapsto t(I,E)])} = T \) iff for some \( d \in \text{dom}(I) \),

\[
\alpha^{(I,E[x \mapsto t(I,E)][y \mapsto d])} = T.
\]

Likewise, we have \( (\exists y \alpha[t/x])^{(I,E)} = T \) iff for some \( d \in \text{dom}(I) \),

\[
(\alpha[t/x])^{(I,E[y \mapsto d])} = T.
\]

In the first case, \( t \) is evaluated under environment \( E \). In the second, \( t \) is evaluated under environment \( E[y \mapsto d] \) (or a further modification).

If \( y \) is free in \( t \), the difference matters!

Also, we have a problem if “\( y \)” is the same variable as \( x \).
Recall the Definition of Substitution

The definition of substitution included the following.

For a variable $x$ and a term $t$:

4. If $\alpha$ is $(Qx \beta)$, then $\alpha[t/x]$ is $\alpha$.

5. If $\alpha$ is $(Qy \beta)$ for some other variable $y$, then
   - If $y$ does not occur in $t$, then $\alpha[t/x]$ is $(Qy \beta[t/x])$.
   - Otherwise, let $z$ be a variable that occurs in neither $\alpha$ nor $t$; then $\alpha[t/x]$ is $(Qz (\beta[z/y])[t/x])$.

With this definition, we always get that, as required,

$$(Qy \alpha)(I,E[x \mapsto t(I,E)]) = ((Qy \alpha)[t/x])^{(I,E)}.$$ 

Proof left to you.

(Note: $((Qy \alpha)[t/x])^{(I,E)}$ may differ from $(Qy(\alpha[t/x]))^{(I,E)}$.)
Rule \( \forall i \) Is Sound

To further clarify the rule \( \forall i \), we show that it is sound. That is,

Suppose that \( \Sigma \models \alpha[y/x] \) and \( y \) is not free in \( \Sigma \) or \( \alpha \).

Then \( \Sigma \models \forall x \alpha \).

**Proof:** Fix an arbitrary \( I \) and \( E \) with \( I \models_E \Sigma \).

The supposition \( \Sigma \models \alpha[y/x] \) thus requires \( I \models_E \alpha[y/x] \).

We need to show that \( I \models_{E[x \mapsto a]} \alpha \) for every \( a \in \text{dom}(I) \).

Consider an arbitrary \( a \in \text{dom}(I) \).

Since \( y \) is not free in \( \Sigma \), the Relevance Lemma yields \( I \models_{E[y \mapsto a]} \Sigma \).

Since \( y \) is not free in \( \alpha \), we have \( \alpha[y/x](I,E[y \mapsto a]) = \alpha(I,E[x \mapsto a]) \).

Therefore \( I \models_{E[x \mapsto a]} \alpha \) for every \( a \), and thus \( I \models_E \forall x \alpha \) as required.
The rule $\exists e$ is sound. That is,

Suppose that $\Sigma, \alpha[u/x] \vdash \beta$ and $u$ is not free in $\Sigma, \alpha$, or $\beta$.

Then $\Sigma, \exists x \alpha \vdash \beta$.

**Proof:** Exercise. Follow the proof of soundness of $\forall i$. 
Completeness of ND for Predicate Logic: Getting started

To show: if \( \Sigma \not\models \alpha \), then \( \Sigma \not\models \alpha \).

**Lemma I:** If \( \Sigma \not\models \alpha \), then \( \Sigma \cup \{\neg \alpha\} \not\models \alpha \).

By rule \( \rightarrow i \), if \( \Sigma \cup \{\neg \alpha\} \vdash \alpha \), then \( \Sigma \vdash \neg \alpha \rightarrow \alpha \). Thus \( \Sigma \vdash \alpha \).

**Lemma II:** If there are \( \mathcal{I} \) and \( E \) s.t. \( \mathcal{I} \models E \Sigma \cup \{\neg \alpha\} \), then \( \Sigma \not\models \alpha \).

\( \mathcal{I} \) and \( E \) satisfy \( \Sigma \) but not \( \alpha \).

**Lemma III** (the big one):

If \( \Sigma \cup \{\neg \alpha\} \not\models \alpha \), then there are \( \mathcal{I} \) and \( E \) such that \( \mathcal{I} \models E \Sigma \cup \{\neg \alpha\} \).
Whence a Domain?

Given: $\Sigma \cup \{\neg \alpha\} \not\vdash \alpha$.

Required: interpretation $\mathcal{H}$ and environment $E$ that satisfy $\Sigma \cup \{\neg \alpha\}$.

To start, we need a domain. Where can we get one?
Whence a Domain?

Given: \( \Sigma \cup \{\neg \alpha\} \not\models \alpha \).

Required: interpretation \( \mathcal{H} \) and environment \( E \) that satisfy \( \Sigma \cup \{\neg \alpha\} \).

To start, we need a domain. Where can we get one?

We use the set of terms. That is, let the domain be

\[ \{ \llbracket t \rrbracket | t \text{ is a term} \} \cdot \]

(The notation ‘\( \llbracket \cdot \rrbracket \)’ indicates that we refer to the domain element, rather than to the expression.)
Interpretation of Terms

For a set $\Sigma$ of premises, we want an interpretation $I$ and an environment $E$, over the domain of terms.

Constants, variables, and functions are easy to handle.

- For a constant symbol $c$, we define $c^I = \llbracket c \rrbracket$.
- For a variable $x$, we define $x^E = \llbracket x \rrbracket$.
- For a $k$-ary function symbol $f$, we define $f^I(\llbracket t_1 \rrbracket, \ldots, \llbracket t_k \rrbracket) = \llbracket f(t_1, \ldots, t_k) \rrbracket$.

Relations pose a problem, since they depend on $\Sigma$. For a relation symbol $R^{(k)}$, we must determine, for each tuple $\langle t_1, \ldots, t_k \rangle$, whether to put $\langle \llbracket t_1 \rrbracket, \ldots, \llbracket t_k \rrbracket \rangle$ into the set $R^I$.

The basic idea is to consider each possible formula, one by one. For each, restrict the set of interpretations to account for it. Since we assume no proof of contradiction exists, we can guarantee that we always have some interpretations left as possible.
As one part of the construction, we require a list of all possible formulas. Since we may have arbitrarily many constant, variable, functions and relation symbols, of any arity, we must take care that everything gets onto the list at some point. For example, if we take the $i$th formula to be $R(c_i)$, then many formulas ($R(x)$, $Q_4(f_7(y_{66}))$, etc.) never appear on the list.

We do the listing “in stages”, starting from stage 1. At stage $j$, consider the first $j$ constants, variables, and function symbols. Form all terms that combine these, using at most $j$ applications of a function. Apply each of the first $j$ relation symbols to each of these terms, and then form all formulas from these that use at most $j$ connectives or quantifiers.

The set of formulas formed this way is large, but finite. After all have been listed, continue to stage $j + 1$. 

---

Soundness and Completeness

Sketch of Proof of Completeness