First-Order 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:

\[ \text{Not all birds can fly.} \]

\[ \text{Every student is younger than some instructor.} \]

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.
- Relations. Properties of objects, alone or in combination.
- Functions. Association of objects to others.
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: \( \forall x \left( \exists y \left( y < x \right) \right) \).
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: \( \{B_1, B_2, B_3\} \).

We can describe the world with relations: 
\( On, OnTable, Red, Blue \).
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:

- \( OnTable(B_1), OnTable(B_2), \neg OnTable(B_3) \).
- \( On: \begin{array}{ccc}
B_1 & B_2 & B_3 \\
B_1 & F & F & F \\
B_2 & F & F & F \\
B_3 & T & F & F \\
\end{array} \)
- 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” \((\forall x (\neg On(x, x)))\).

“A box on the table is not on any box”:
\[
\left(\forall x \left( OnTable(x) \rightarrow (\neg (\exists y \ On(x, y)))\right)\right).
\]

Some properties depend on the situation.

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

“Some box is on a box that is on the table”:
\[
\left(\exists x \ \exists y \left( On(x, y) \land OnTable(y)\right)\right).
\]
A graph is a binary relation.

A finite graph:

\[
\begin{array}{c}
\text{Nodes: } \{1, 2, 3, 4\} \\
\text{Edges: } \{(1, 2), (2, 3), (3, 4), (4, 1)\}
\end{array}
\]

An infinite graph:

\[
\begin{array}{c}
\text{Points on line: } \{(x, y) \mid y = 1 + x/2\}
\end{array}
\]

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

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

holds.
A relational database is a listing of one or more 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 relations.

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

$$\exists x \left( \text{Person}(x) \land \neg \exists y \left( \text{NumberOf}(x, y) \right) \right).$$
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.”

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

“Whenever \( x \) has some phone number \( y \), then \( x \) is a person.”

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

These two formulas are respectively equivalent to

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

and to

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

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 *predicate*, or *relation*, represents a property that an individual, or collection of individuals, may (or may not) have. In English, we might express a predicate as

“______ 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 “$S(Alex)$”.

Similarly, we might use $I(Sam)$ for “Sam is an instructor” and $Y(Alex,Sam)$ for “Alex is younger than Sam”.
Mathematically, we represent a relation 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 relation is that it must be a subset of the domain.

A $k$-ary relation is a set of $k$-tuples of domain elements. For example, the binary relation 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 $On$ relation 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 relation “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 ‘∀’, the *universal quantifier* symbol, and “there exists” is denoted by ‘∃’, the *existential quantifier* symbol.

In FOL, the above comes out as the formula

$$
\forall \varepsilon \left( (\varepsilon > 0) \rightarrow \exists \delta \left( (\delta > 0) \land \forall y \left( (|x - y| < \delta) \rightarrow (|f(x) - f(y)| < \varepsilon) \right) \right) \right)
$$
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 First-Order 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 FOL, 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 $n = 2$ (a binary function symbol), 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 \( +\left(x, s(y)\right)\).

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 \( f\left(g(a, f(a))\right) \) are terms.

The expressions \( g(a) \) and \( f(f(a), a) \) are not terms.
Atomic Formulas

As in propositional logic, a formula represents a proposition (a true/false statement). The relation 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 relation 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 $(t_1 P t_2)$. 
General Formulas

We define the set of well-formed formulas of first-order 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 FOL formulas are similar to parse trees for propositional formulas.

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

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

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

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

\[
\forall x (F(b) \rightarrow \exists y \forall z (G(y, z) \lor H(u, x, y)))
\]
Semantics: Interpretations

We cover more on syntax later, but we first start the discussion of semantics.

**Definition:** Fix a set $\mathcal{L}$ of constant symbols, function symbols, and relation symbols. (The “language” of our formulas.)

An *interpretation* $\mathcal{I}$ (for the set $\mathcal{L}$) consists of

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

Huth and Ryan use the term “*model*” instead of “interpretation.” (Not a standard usage.)
Values of Variable-Free Terms

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 $\mathcal{I}$. For each term $t$ containing no variables, the value of $t$ under interpretation $\mathcal{I}$, denoted $t^\mathcal{I}$, is as follows.

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

The value of a term is always a member of the domain of $\mathcal{I}$. 

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

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

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

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

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

Consider an interpretation \( \mathcal{I} \) with

\[
\begin{align*}
\text{Domain: } & \mathbb{N}, \text{the natural numbers} \\
0^\mathcal{I}: & \text{zero} \\
f^\mathcal{I}: & \text{successor; } \{ \langle x, x + 1 \rangle \mid x \in \mathbb{N} \} \\
E^\mathcal{I}: & \text{“is even”; } \{ 2y \mid y \in \mathbb{N} \}
\end{align*}
\]

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

Formula \( E(f(0)) \) means “1 is even”, and \( E(f(0))^\mathcal{I} = \text{F} \).
Formula \( E(f(f(0))) \) means “2 is even”, and \( E(f(f(0)))^\mathcal{I} = \text{T} \).

What about some other interpretation?
Example, Continued

Let $\mathcal{J}$ be the interpretation with

Domain: $\mathbb{Q}$, the rational numbers

$0^\mathcal{J}$: two

$f^\mathcal{J}$: halving; $\{ \langle x, x/2 \rangle \mid x \in \mathbb{Q} \}$

$E^\mathcal{J}$: “is an integer”; $\{ x \mid x \in \mathbb{Z} \}$

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

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

**Exercise:** in both $\mathcal{I}$ and $\mathcal{J}$, the formula $\left( E(f(f(0))) \land E(f(0)) \right)$ receives value F. Find another interpretation which gives it the value T.
Two often-overlooked points about interpretations.

1. There is NO default meaning for relation, 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 relation 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 $\mathbb{N}$. 
To discuss the evaluation of formulas that contain variables, we need a few more concepts from syntax.

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.)
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 $(\beta \star \gamma)$, 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 closed formula, or a 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. 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.
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 $(P(x) \land (\exists x \ Q(x)))$. 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 $(\forall x (\exists y ((x+y) = z)))$?
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. ...
Substitution—Formal Definition (2)

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

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 $(\forall x \left( \exists y \beta \right))$.
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 $\left( \forall x \left( \exists y \beta \right) \right)$.

Select a new variable, say $w$. Then

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

and

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

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

$$\left( \forall x \exists w((x + w) = (y - 1)) \right).$$
Semantics of Predicate Logic
In propositional logic, semantics was described in terms of valuations to propositional atoms.

FOL includes more ingredients (i.e., predicates, functions, variables, terms, constants, etc.) and, hence, the semantics for FOL 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, relations and functions.

Formulas that include variables, and perhaps quantifiers, require additional information, known as an environment (or assignment).
A first-order *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 $\mathcal{I}$ be the interpretation with domain $\mathbb{N}$, $c^\mathcal{I} = 2$ and $P^\mathcal{I} = “is even”$. Then $\alpha_1^\mathcal{I} = T$, but $\alpha_2^\mathcal{I}$ is undefined.

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

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 $\alpha_2^{(\mathcal{I},E)}$. 
The combination of an interpretation and an environment supplies a value for every term.

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

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

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

But first, a few examples.
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 \(\left(s(s(0) + s(x))\right)^{(I,E)} = 6\) and \(\left(s(x + s(x + s(0)))\right)^{(I,E)} = 9\).
Meaning of Terms—Example 2

- \( \text{dom}\{\mathcal{J}\} \) is the collection of all words over the alphabet \( \{a, b\} \),
  \( 0^\mathcal{J} = a \),
  \( s^\mathcal{J} \) appends \( a \) to the end of a string, and
  \( +^\mathcal{J} \) is concatenation.

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

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

and

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

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

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

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 $\text{dom}(\mathcal{I}) = \{a, b\}$ and $R^\mathcal{I} = \{\langle a, a \rangle, \langle a, b \rangle, \langle b, b \rangle\}$.

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

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

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

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

Since \( \langle b, a \rangle \notin R^\mathcal{I} \), we have

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

and thus

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

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

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

Since \( \langle b, a \rangle \not\in R^I \), we have

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

and thus

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

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

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

Why did we not write simply

$$R(b, a) = F$$

or perhaps

$$R(b, a)^{(I, E)} = F?$$
In the previous example, we wrote

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

Why did we not write simply

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

or perhaps

\[ R(b, a)^{(I, E)} = F? \]

Because “\( R(b, a) \)” 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 $\mathcal{I}$ and environment $E$ **satisfy** a formula $\alpha$, denoted $\mathcal{I} \models_E \alpha$, if $\alpha^{(\mathcal{I},E)} = T$; they do not satisfy $\alpha$, denoted $\mathcal{I} \not\models_E \alpha$, if $\alpha^{(\mathcal{I},E)} = F$.

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

If $\mathcal{I} \models_E \alpha$ for every $E$, then $\mathcal{I}$ **satisfies** $\alpha$, denoted $\mathcal{I} \models \alpha$. 
Example. Consider the formula \((\exists y \mathcal{R}(x, y \oplus y))\).
(For \(\mathcal{R}\) a binary relation and \(\oplus\) a binary function.)

Suppose \(\text{dom}(\mathcal{I}) = \{1, 2, 3, \ldots\}\),
\(\oplus^\mathcal{I}\) is the addition operation, and
\(\mathcal{R}^\mathcal{I}\) is the equality relation.

Then \(\mathcal{I} \models_{E} (\exists y \mathcal{R}(x, y \oplus y))\) 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: Satisfiability 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 $x$, $R(x)$ iff $R(a_1)$ and ... and $R(a_k)$

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

where $R$ is a property.
Lemma:

Let $\alpha$ be a first-order formula, $\mathcal{I}$ be an interpretation, and $E_1$ 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

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

Proof by induction on the structure of $\alpha$. 
Suppose $\Sigma$ is a set of formulas and $\alpha$ is a formula. We say that $\alpha$ is a logical consequence of $\Sigma$, written as $\Sigma \models \alpha$, iff for any interpretation $\mathcal{I}$ and environment $E$, we have $\mathcal{I} \models_E \Sigma$ implies $\mathcal{I} \models_E \alpha$.

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

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

Proof by contradiction. Suppose there are $I$ and $E$ such that $I \not\models_E (\forall x (\alpha \rightarrow \beta)) \rightarrow ((\forall x \alpha) \rightarrow (\forall x \beta))$.

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 $I \not\models_E (\forall x \alpha)$ and $I \models_E (\forall x \beta)$.

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 II: Entailment

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

\[\text{for every } a \in \text{dom}(I), I \models_{E[x \mapsto a]} (\neg \gamma).\]

Again by definition (for a formula with \(\neg\)), this is equivalent to

\[\text{for every } a \in \text{dom}(I), I \not\models_{E[x \mapsto a]} \gamma\]

and also

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

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

\[ ((\forall x \alpha) \rightarrow (\forall x \beta)) \n\equiv (\forall x (\alpha \rightarrow \beta)) \]

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

\[ ((\forall x \alpha) \rightarrow (\forall x \beta)) \not\models (\forall x(\alpha \rightarrow \beta)) \ . \]

(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 \( R(x) \). Let \( \mathcal{I} \) have domain \{a, b\} and \( R^\mathcal{I} = \{a\} \). Then
\[ \mathcal{I} \models (\forall x \alpha) \rightarrow (\forall x \beta) \] for any \( \beta \). (Why?)

To obtain \( \mathcal{I} \not\models \forall x(\alpha \rightarrow \beta) \), we can use \( \neg R(x) \) for \( \beta \). (Why?)

Thus \( ((\forall x \alpha) \rightarrow (\forall x \beta)) \not\models (\forall x(\alpha \rightarrow \beta)) \), as required. (Why?)
Example: for any formula \( \alpha \) and term \( t \),

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

Recall that functions must be total!