# Back to the knights and knaves island

During your Thanksgiving trip, you found another island with knights and knaves (you have a feeling that you may encounter many such islands this semester). Knights always tell the truth and knaves always lie.

You talk to every person on the island and each one says "All of us on the island are of the same type".

Are they really the same type? Can you determine which type?

a. They are all knights.
b. They are all knaves.
c. Some are knights and some are knaves.

# *Predicate Logic: Informal Introduction and Translation*

Alice Gao

October 12, 2017

These slides are based on materials from CS245 at UWaterloo and CPSC121 at UBC.

# What can't we express using propositional logic?

Can we express the following ideas using propositional logic?

- Translate this sentence: Alice is married to Jay and Alice is not married to Leon.
- Translate this sentence: Every bear likes honey.
- Define what it means for a natural number to be prime.

# What can't we express using propositional logic?

A few things that are difficult to express using propositional logic:

- Relationships among individuals: Alice is married to Jay and Alice is not married to Leon.
- Generalizing patterns: Every bear likes honey.
- Infinite domains: Define what it means for a natural number to be prime.

We can use predicate logic (first-order logic) to express all of these.

# Would you really use predicate logic?

Examples of predicate logic in CS245 so far:

1. *Every* well-formed formula has an equal number of left and right brackets.
2. For *every* truth valuation t, if all the premises are true under t, then the conclusion is true under t.
3. If *there does not exist* a natural deduction proof from the premises to the conclusion, then the premises do not semantically entail the conclusion.

# Would you really use predicate logic?

Examples of predicate logic in Computer Science:

1. Data Structure: *Every* key stored in the left subtree of a node $N$ is smaller than the key stored at $N$.
2. Algorithms: in the worst case, *every* comparison sort requires at least $cn \log n$ comparisons to sort $n$ values, *for some* constant $c > 0$.
3. Java example: *there is no* path via references from any variable in scope to any meomry location available for garbage collection...
4. Database query: select a person whose age is greater than or equal to the age of *every* other person in the table.

# Elements of predicate logic

Predicate logic generalizes propositional logic.

New things in predicate logic:

- Domains
- Predicates
- Quantifiers

# Domains

A *domain* is a non-empty set of objects. It is a world that our statement is situated within.

Examples of domains: natural numbers, people, animals, etc.

Why is it important to specify a domain? *The same statement can have different truth values in different domains.*

Consider this statement: There exists a number whose square is $2$.

- If our domain is the set of natural numbers, is this statement true or false?
- If our domain is the set of real numbers, is this statement true or false?

# Objects in a domain

*Constants*: concrete objects in the domain

- Natural numbers: 0, 6, 100, ...
- Alice, Bob, Eve, ...
- Animals: Winnie the Pooh, Micky Mouse, Simba, ...

*Variables*: placeholders for concrete objects, e.g. x, y, z.

A variable lets us refer to an object without specifying which particular object it is.

# Predicates

A *predicate* represents

- a property of an individual, or
- a relationship among mulitple individuals.

Think of a predicate as a special type of function which takes constants and/or variables as inputs and *outputs* `T`/`F`. It can have any number of arguments.

Examples:

- Define $L(x)$ to mean "x is a lecturer". (unary predicate)
  - Alice is a lecturer: $L(Alice)$
  - Micky Mouse is not a lecturer: $(\neg L(MickyMouse))$
  - $y$ is a lecturer: $L(y)$
- Define $Y(x, y)$ to mean "x is younger than y". (binary predicate/relation)
  - Alex is younger than Sam: $Y(Alex, Sam)$
  - $a$ is younger than $b$: $Y(a, b)$

# Quantifiers

For how many objects in the domain is the statement true?

- The universal quantifier $\forall$: the statement is true for every object in the domain.
- The existential quantifier $\exists$: the statement is true for one or more objects in the domain.

# Translating English into Predicate Logic

Let the domain be the set of animals. $\mathit{Honey}(x)$ means that $x$ likes honey. $\mathit{Bear}(x)$ means that $x$ is a bear.

Consider the following translations of English sentences into predicate logic formulas. Are the translations correct?

1. At least one animal likes honey. $(\exists x\ \mathit{Honey}(x))$.
2. Not every animal likes honey. $(\neg(\forall x\ \mathit{Honey}(x)))$.

a. Both are correct.

b. 1 is correct and 2 is wrong.

c. 1 is wrong and 2 is correct.

d. Both are wrong.

# Translating English into Predicate Logic

Translate the following sentences into predicate logic.

1. All animals like honey.
2. At least one animal likes honey.
3. Not every animal likes honey.
4. No animal likes honey.

Let the domain be the set of animals. $\mathit{Honey}(x)$ means that $x$ likes honey. $\mathit{Bear}(x)$ means that $x$ is a bear.

# Translating English into Predicate Logic

Translate the following sentences into predicate logic.

5. No animal dislikes honey.
6. Not every animal dislikes honey.
7. Some animal dislikes honey.
8. Every animal dislikes honey.

Let the domain be the set of animals. *Honey*$(x)$ means that $x$ likes honey. *Bear*$(x)$ means that $x$ is a bear.

# Translating English into Predicate Logic

Consider this sentence: *every* bear likes honey. Which one is the correct translation into predicate logic?

a. $(\forall x\,(Bear(x) \wedge Honey(x)))$

b. $(\forall x\,(Bear(x) \vee Honey(x)))$

c. $(\forall x\,(Bear(x) \rightarrow Honey(x)))$

d. $(\forall x\,(Honey(x) \rightarrow Bear(x)))$

Let the domain be the set of animals. *Honey*$(x)$ means that $x$ likes honey. *Bear*$(x)$ means that $x$ is a bear.

# Translating English into Predicate Logic

Consider this sentence: *some* bear likes honey. Which one is the correct translation into predicate logic?

a. $(\exists x\,(Bear(x) \wedge Honey(x)))$

b. $(\exists x\,(Bear(x) \vee Honey(x)))$

c. $(\exists x\,(Bear(x) \rightarrow Honey(x)))$

d. $(\exists x\,(Honey(x) \rightarrow Bear(x)))$

Let the domain be the set of animals. *Honey*$(x)$ means that $x$ likes honey. *Bear*$(x)$ means that $x$ is a bear.

# Multiple Quantifiers

Let the domain be the set of people. Let $L(x, y)$ mean that person $x$ likes person $y$.

Translate the following formulas into English.

1. $(\forall x\,(\forall y\ L(x, y)))$
2. $(\exists x\,(\exists y\ L(x, y)))$
3. $(\forall x\,(\exists y\ L(x, y)))$
4. $(\exists y\,(\forall x\ L(x, y)))$

# 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. | $P$, $Q$, ...$P_1$, $P_2$, ..., $Q_1$, $Q_2$, ... |
| 5. | Connectives: | $\neg$, $\wedge$, $\vee$, $\rightarrow$, 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.

# Terms

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 $n = 2$ (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 $+(x, s(y))$.

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 $f\big(g(a, f(a))\big)$ 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 Predicate Logic formulas are similar to parse trees for propositional formulas.

- Quantifiers $\forall x$ and $\exists y$ form nodes is 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: $(\forall x\,((P(x) \to Q(x)) \wedge S(x, y)))$.

# Examples: Parse trees

Example: $(\forall x\,((P(x) \to Q(x)) \wedge S(x, y)))$.

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

# Examples: Parse trees

Example: $(\forall x\,((P(x) \to Q(x)) \wedge S(x,y)))$.

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

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

$$\forall x\,\big(F(b) \to \exists y\ \forall z\,\big(G(y,z) \vee H(u,x,y)\big)\big)\ .$$

# Semantics: Interpretations

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

**Definition:** Fix a set $L$ of constant symbols, function symbols, and relation symbols. (The "language" of our formulas.)

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

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

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 $f(t_1, \ldots, t_n)$, the value $t^{\mathcal{I}}$ is $f^{\mathcal{I}}(t_1^{\mathcal{I}}, \ldots, t_n^{\mathcal{I}})$.

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

# Formulas with Variable-Free Terms

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

**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

$$\alpha^{\mathcal{I}} = \begin{cases} \mathtt{T} & \text{if } \langle t_1^{\mathcal{I}}, \ldots, t_n^{\mathcal{I}} \rangle \in R^{\mathcal{I}} \\ \mathtt{F} & \text{otherwise.} \end{cases}$$

- 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 $E\big(f(0)\big)$ and $E\big(f\big(f(0)\big)\big)$ are both formulas.

Consider an interpretation $\mathcal{J}$ with

| | |
|---|---|
| Domain: | $\mathbb{N}$, the natural numbers |
| $0^{\mathcal{J}}$: | zero |
| $f^{\mathcal{J}}$: | successor; $\{\, \langle x, x+1 \rangle \mid x \in \mathbb{N} \,\}$ |
| $E^{\mathcal{J}}$: | "is even"; $\{\, 2y \mid y \in \mathbb{N} \,\}$ |

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

Formula $E\big(f(0)\big)$ means "1 is even", and $E\big(f(0)\big)^{\mathcal{J}} = \mathtt{F}$.
Formula $E\big(f\big(f(0)\big)\big)$ means "2 is even", and $E\big(f\big(f(0)\big)\big)^{\mathcal{J}} = \mathtt{T}$.

What about some other interpretation?

# Example, Continued

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

| | |
|---|---|
| Domain: | $\mathbb{R}$, the rational numbers |
| $0^{\mathcal{J}}$: | two |
| $f^{\mathcal{J}}$: | halving; $\{ \langle x, x/2 \rangle \mid x \in \mathbb{R} \}$ |
| $E^{\mathcal{J}}$: | "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 $\Big(E(f(f(0))) \wedge E(f(0))\Big)$ receives value `F`. Find another interpretation which gives it the value `T`.

# "Gotchas"

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 $\mathbb{N}$.

# Variables

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 $\Big(\forall x\ \big(\exists y\ (x + y = z)\big)\Big)$, $x$ is bound (by $\forall x$), $y$ is bound (by $\exists y$), and $z$ is free.

*Example.* In formula $\Big(P(x) \wedge \big(\forall x\,(\neg Q(x))\big)\Big)$, 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 $\mathsf{FV}(\alpha)$ defined as follows.

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

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 $E\big(f(f(x))\big)$.
- $E(f(x+y))[y/x]$ is $E(f(y+y))$.

Substitution does NOT affect bound occurrences of the variable.

- If $\beta$ is $\big(\forall x\,(E(f(x)) \wedge S(x,y))\big)$, then $\beta[g(x,y)/x]$ is $\beta$, because $\beta$ has no free occurrence of $x$.

# Examples: Substitution

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

# Examples: Substitution

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

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

# Examples: Substitution

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

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

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

# Examples: Substitution

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

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

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

At first thought, we might say $\Big(\forall x\,\big(\exists y\,((x+y)=(y-1))\big)\Big)$. 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 $\alpha = S(x) \wedge \forall y\,(P(x) \to Q(y))$; 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.

# Substitution—Formal Definition

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 $P\Big(t_1[t/x], \ldots, t_k[t/x]\Big)$.
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. ...

*(Continued next page...)*

# Substitution—Formal Definition (2)

For variable $x$, term $t$ and formula $\alpha$:

$\vdots$

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 $\big(\forall x\,(\exists y\,(x+y=z))\big)$, 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 $\big(\forall x\,(\exists y\ \beta)\big)$.

# Example, Revisited

*Example.* If $\alpha$ is $\big(\forall x\,(\exists y\,(x+y=z))\big)$, 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 $\big(\forall x\,(\exists y\ \beta)\big)$.

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

$$\big(\forall x\ \exists w((x+w)=(y-1))\big)\ .$$

# Semantics of Predicate Logic

# 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, relations and functions.

Formulas that include variables, and perhaps quantifiers, require additional information, known as an *environment* (or *assignment*).

# Environments

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.

# Constants Vs. Variables

*Example*: Let $\alpha_1$ be $P(c)$ (where $c$ is a constant), and let $\alpha_2$ be $P(x)$ (where $x$ a variable).

Let $\mathcal{J}$ be the interpretation with domain $\mathbb{N}$, $c^{\mathcal{J}} = 2$ and $P^{\mathcal{J}} =$ "is even". Then $\alpha_1^{\mathcal{J}} = \mathtt{T}$, but $\alpha_2^{\mathcal{J}}$ is undefined.

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

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 $\alpha_2^{(\mathcal{J},E)}$.

# Meaning of Terms

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.

# 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 $s\big(s(0) + s(x)\big)$ and $s\Big(x + s\big(x + s(0)\big)\Big)$ are both terms.

The following are examples of interpretations and environments.

- $dom\{\mathcal{J}\} = \{0, 1, 2, ...\}$, $0^{\mathcal{J}} = 0$, $s^{\mathcal{J}}$ is the successor function and $+^{\mathcal{J}}$ is the addition operation. Then, if $E(x) = 3$, the terms get values $\big(s\big(s(0) + s(x)\big)\big)^{(\mathcal{J},E)} = 6$ and $\Big(s\big(x + s\big(x + s(0)\big)\big)\Big)^{(\mathcal{J},E)} = 9$.

# Meaning of Terms—Example 2

- $dom\{\mathcal{J}\}$ 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

$$\big(s(s(0) + s(x))\big)^{(\mathcal{J},E)} = aaabaaa$$

  and

$$\Big(s\big(x + s(x + s(0))\big)\Big)^{(\mathcal{J},E)} = abaabaaaaa \ .$$

# 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 $\mathrm{d}$", denoted $E[x \mapsto \mathsf{d}]$,
is given by

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

# Values of Quantified Formulas

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

- $(\forall x\, \alpha)^{(\mathcal{J},E)} = \begin{cases} \mathtt{T} & \text{if } \alpha^{(\mathcal{J},E[x\mapsto \mathsf{d}])} = \mathtt{T} \text{ for every d in } dom(\mathcal{J}) \\ \mathtt{F} & \text{otherwise} \end{cases}$
- $(\exists x\, \alpha)^{(\mathcal{J},E)} = \begin{cases} \mathtt{T} & \text{if } \alpha^{(\mathcal{J},E[x\mapsto \mathsf{d}])} = \mathtt{T} \text{ for some d in } dom(\mathcal{J}) \\ \mathtt{F} & \text{otherwise} \end{cases}$

Note: The values of $(\forall x\, \alpha)^{(\mathcal{J},E)}$ and $(\exists x\, \alpha)^{(\mathcal{J},E)}$ do not depend on the value of $E(x)$.

The value $E(x)$ only matters for free occurrences of $x$.

# Examples: Value of a Quantified Formula

*Example.* Let $dom(\mathcal{J}) = \{\mathsf{a}, \mathsf{b}\}$ and $R^{\mathcal{J}} = \{\langle \mathsf{a}, \mathsf{a}\rangle, \langle \mathsf{a}, \mathsf{b}\rangle, \langle \mathsf{b}, \mathsf{b}\rangle\}$.

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

- $R(x, x)^{(\mathcal{J}, E)} = \mathtt{T}$, since $\langle E(x), E(x)\rangle = \langle \mathsf{a}, \mathsf{a}\rangle \in R^{\mathcal{J}}$.
- $R(y, x)^{(\mathcal{J}, E)} = \mathtt{F}$, since $\langle E(y), E(x)\rangle = \langle \mathsf{b}, \mathsf{a}\rangle \notin R^{\mathcal{J}}$.
- $\big(\exists y\, R(y, x)\big)^{(\mathcal{J}, E)} = \mathtt{T}$, since $R(y, x)^{(\mathcal{J}, E[y \mapsto \mathsf{a}])} = \mathtt{T}$.

  (That is, $\langle E[y \mapsto \mathsf{a}](y), E[y \mapsto \mathsf{a}](x)\rangle = \langle \mathsf{a}, \mathsf{a}\rangle \in R^{\mathcal{J}}$).
- What is $\big(\forall x\, (\forall y\, R(x, y))\big)^{(\mathcal{J}, E)}$?

# Examples: Continued

*Example.* Let $dom(\mathcal{J}) = \{\mathsf{a}, \mathsf{b}\}$ and $R^{\mathcal{J}} = \{\langle \mathsf{a}, \mathsf{a}\rangle, \langle \mathsf{a}, \mathsf{b}\rangle, \langle \mathsf{b}, \mathsf{b}\rangle\}$.

Let $E(x) = \mathsf{a}$ and $E(y) = \mathsf{b}$.

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

  Since $\langle \mathsf{b}, \mathsf{a}\rangle \notin R^{\mathcal{J}}$, we have

  $$R(x, y)^{(\mathcal{J}, E[x \mapsto \mathsf{b}][y \mapsto \mathsf{a}])} = \mathtt{F} \ ,$$

  and thus

  $$\big(\forall x \, (\forall y \, R(x, y))\big)^{(\mathcal{J}, E)} = \mathtt{F} \ .$$

# Examples: Continued

*Example.* Let $dom(\mathcal{J}) = \{\mathsf{a}, \mathsf{b}\}$ and $R^{\mathcal{J}} = \{\langle \mathsf{a}, \mathsf{a}\rangle, \langle \mathsf{a}, \mathsf{b}\rangle, \langle \mathsf{b}, \mathsf{b}\rangle\}$.

Let $E(x) = \mathsf{a}$ and $E(y) = \mathsf{b}$.

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

  Since $\langle \mathsf{b}, \mathsf{a}\rangle \notin R^{\mathcal{J}}$, we have

  $$R(x, y)^{(\mathcal{J}, E[x \mapsto \mathsf{b}][y \mapsto \mathsf{a}])} = \mathtt{F} \ ,$$

  and thus

  $$\big(\forall x \, (\forall y \, R(x, y))\big)^{(\mathcal{J}, E)} = \mathtt{F} \ .$$

- What about $\big(\forall x \, (\exists y \, R(x, y))\big)^{(\mathcal{J}, E)}$?

# A Question of Syntax

In the previous example, we wrote

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

Why did we not write simply

$$R(\mathsf{b}, \mathsf{a}) = \mathtt{F}$$

or perhaps

$$R(\mathsf{b}, \mathsf{a})^{(\mathcal{J}, E)} = \mathtt{F} \ ?$$

# A Question of Syntax

In the previous example, we wrote

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

Why did we not write simply

$$R(\mathsf{b},\mathsf{a}) = \mathtt{F}$$

or perhaps

$$R(\mathsf{b},\mathsf{a})^{(\mathcal{J},E)} = \mathtt{F} \ ?$$

Because "$R(\mathsf{b},\mathsf{a})$" is not a formula. The elements a and b of $dom(\mathcal{J})$ are not symbols in the language; they cannot appear in a formula.

# Satisfaction of Formulas

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

| *Form of $\alpha$* | *Condition for $\mathcal{J} \models_E \alpha$* |
|---|---|
| $R(t_1, \ldots, t_k)$ | $\left\langle t_1^{(\mathcal{J},E)}, \ldots, t_k^{(\mathcal{J},E)} \right\rangle \in R^{\mathcal{J}}$ |
| $(\neg\beta)$ | $\mathcal{J} \not\models_E \beta$ |
| $(\beta \wedge \gamma)$ | both $\mathcal{J} \models_E \beta$ and $\mathcal{J} \models_E \gamma$ |
| $(\beta \vee \gamma)$ | either $\mathcal{J} \models_E \beta$ or $\mathcal{J} \models_E \gamma$ (or both) |
| $(\beta \rightarrow \gamma)$ | either $\mathcal{J} \not\models_E \beta$ or $\mathcal{J} \models_E \gamma$ (or both) |
| $(\forall x\, \beta)$ | for every a $\in dom(\mathcal{J})$, $\mathcal{J} \models_{E[x \mapsto \mathsf{a}]} \beta$ |
| $(\exists x\, \beta)$ | there is some a $\in dom(\mathcal{J})$ such that $\mathcal{J} \models_{E[x \mapsto \mathsf{a}]} \beta$ |

If $\mathcal{J} \models_E \alpha$ for every $E$, then $\mathcal{J}$ *satisfies* $\alpha$, denoted $\mathcal{J} \models \alpha$.

# Example: Satisfaction

*Example.* Consider the formula $(\exists y\, R(x, y \oplus y))$.

(For $R$ a binary relation and $\oplus$ a binary function.)

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

Then $\mathcal{I} \vDash_E (\exists y\, R(x, y \oplus y))$ 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 $\mathcal{I} \vDash_E \alpha$ for every $\mathcal{I}$ and $E$,
- *satisfiable* if some interpretation and environment satisfy $\alpha$; that is, if $\mathcal{I} \vDash_E \alpha$ for some $\mathcal{I}$ and $E$, and
- *unsatisfiable* if no interpretation and environment satisfy $\alpha$; that is, if $\mathcal{I} \nvDash_E \alpha$ for every $\mathcal{I}$ and $E$.

(The term "tautology" is not used in predicate logic.)

# Example: Satisfiability and Validity

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

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

$\alpha$ is not valid. (Why?)

# Quantifiers Over Finite Domains

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:

$$\text{For all } x,\ R(x) \quad \text{iff} \quad R(a_1) \text{ and } \ldots \text{ and } R(a_k)$$

$$\text{There exists } x,\ R(x) \quad \text{iff} \quad R(a_1) \text{ or } \ldots \text{ or } R(a_k)$$

where $R$ is a property.

# Relevance Lemma

**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

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

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

# Logical Consequence

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 $\models \Big((\forall x(\alpha \to \beta)) \to \big((\forall x\, \alpha) \to (\forall x\, \beta)\big)\Big)$.

Proof by contradiction. Suppose there are $\mathcal{I}$ and $E$ such that

$$\mathcal{I} \not\models_E \Big((\forall x(\alpha \to \beta)) \to \big((\forall x\, \alpha) \to (\forall x\, \beta)\big)\Big) \ .$$

Then we must have $\mathcal{I} \models_E (\forall x(\alpha \to \beta))$ and $\mathcal{I} \not\models_E ((\forall x\, \alpha) \to (\forall x\, \beta))$;

the second gives $\mathcal{I} \models_E (\forall x\, \alpha)$ and $\mathcal{I} \not\models_E (\forall x\, \beta)$.

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

Thus $\mathcal{I} \models_E \forall x\, \beta$, a contradiction.

# Example II: Entailment

*Example.* Show that $\big(\forall x(\neg\gamma)\big) \vDash \big(\neg(\exists x\,\gamma)\big)$.

# Example II: Entailment

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

Suppose that $\mathcal{I} \models_E \big(\forall x(\neg\gamma)\big)$. By definition, this means

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

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

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

and also

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

This last is the definition of $\mathcal{I} \models_E \big(\neg(\exists x\,\gamma)\big)$, as required.

# Example

*Example*: Show that, in general,

$$((\forall x\, \alpha) \to (\forall x\, \beta)) \not\models (\forall x(\alpha \to \beta)) \ .$$

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

# Example

*Example*: Show that, in general,

$$((\forall x\, \alpha) \to (\forall x\, \beta)) \not\models (\forall x(\alpha \to \beta)) \ .$$

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

Key idea: $\varphi_1 \to \varphi_2$ yields true whenever $\varphi_1$ is false.

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

# Example

*Example*: Show that, in general,

$$((\forall x\, \alpha) \to (\forall x\, \beta)) \not\models (\forall x(\alpha \to \beta)) \ .$$

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

Key idea: $\varphi_1 \to \varphi_2$ yields true whenever $\varphi_1$ is false.

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

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

Thus $((\forall x\, \alpha) \to (\forall x\, \beta)) \not\models (\forall x(\alpha \to \beta))$, as required. (Why?)

# Example

*Example*: for any formula $\alpha$ and term $t$,

$$\vDash ((\forall x\, \alpha) \to \alpha[t/x]) \ .$$

Recall that functions must be total!