Warm-Up Problem

Review the definition of term [re: binary function] and predicate (atomic formula) from last lecture.
Review lecture 10 at the end when we translate animals and honey (third last slide) and do these translations (we'll do them later in class).
Think about some things you like and dislike about the lectures thus far.
A survey will come at the end of class.

Predicate Logic: More Syntax

Carmen Bruni

Lecture 11

These slides are based on materials from Alice Gao's slides which in turn were based on CS245 at UWaterloo and CPSC121 at UBC.

Based on slides by Jonathan Buss, Lila Kari, Anna Lubiw and Steve Wolfman with thanks to B. Bonakdarpour, A. Gao, D. Maftuleac, C. Roberts, R. Trefler, and P. Van Beek

Learning Goals

- Translate to and from Predicate Logic
- Draw a Parse Tree for a Predicate Logic Formula
- Define free and bound variables and determine when variables in a formula are of which type
- Define a substitution and be able to perform substitutions with or without variable capture in formulas.

The Language of Predicate Logic

The seven kinds of symbols:

1.	Constant symbols.	Usually $c, d, c_1, c_2, \dots, d_1, d_2 \dots$
2.	Variables.	Usually $x, y, z, \dots x_1, x_2, \dots, y_1, y_2 \dots$
3.	Function symbols.	Usually $f, g, h, \dots f_1, f_2, \dots, g_1, g_2,$ ...
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.

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.
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.

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. $(\exists y\,(\forall x\ L(x, y)))$

Parse Trees of Predicate Logic Formulas

New elements in the parse tree:

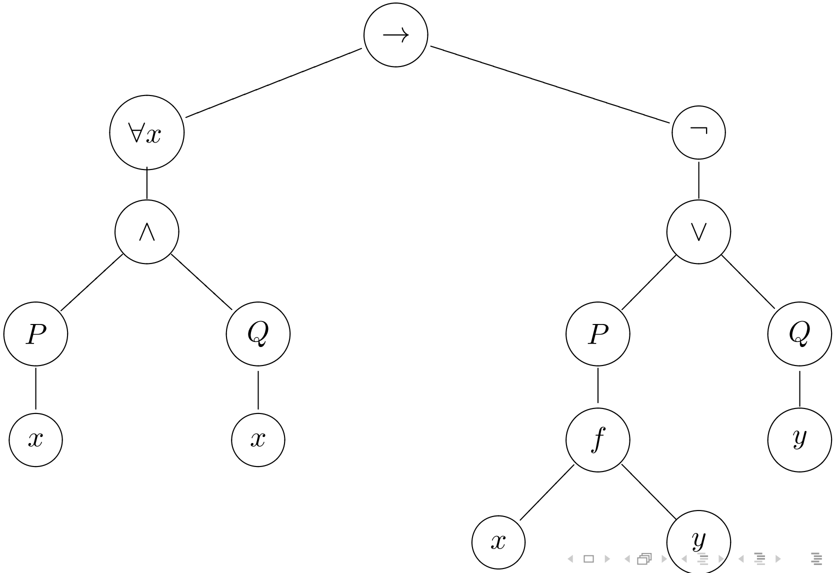
- Quantifiers $\forall x$ and $\exists y$ have one subtree, similar to the unary connective negation.
- A predicate symbol $P(t_1, t_2, \dots, t_n)$ has a node labelled P with a sub-tree for each of the terms $t_1, t_2, \dots, t_n$.
- A function symbol $f(t_1, t_2, \dots, t_n)$ has a node labelled f with a sub-tree for each of the terms $t_1, t_2, \dots, t_n$.

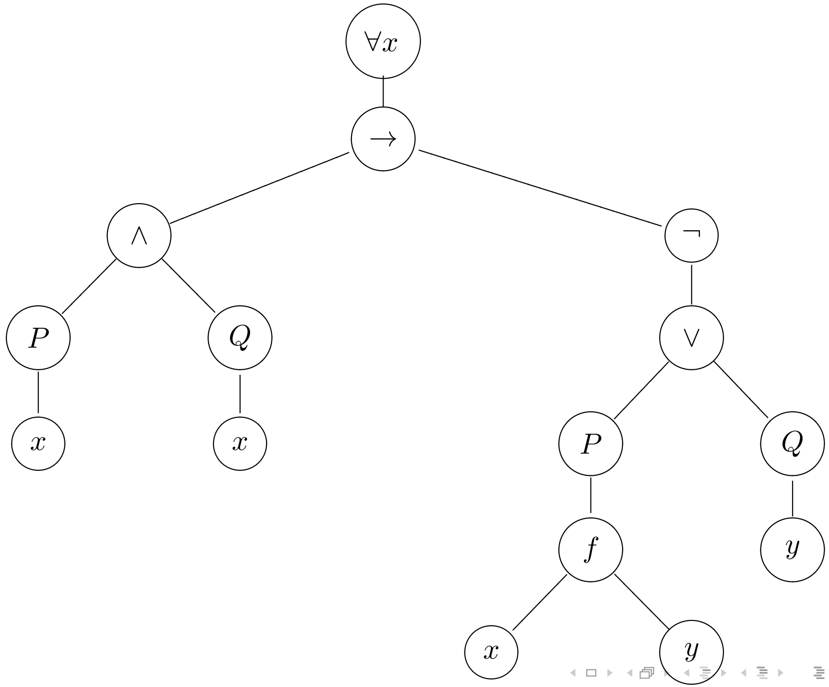
Below P and q are unary predicate symbols and f is a binary function symbol.

Example 1: $((\forall x\,(P(x) \land Q(x))) \to (\neg(P(f(x,y)) \lor Q(y))))$

Example 2: $(\forall x\,((P(x) \land Q(x)) \to (\neg(P(f(x,y)) \lor Q(y)))))$

Example 1 Solution (Example 2 on next slide)




$\forall x$
$\rightarrow$
$\wedge$
$\neg$
P
Q
$\vee$
x
x
P
Q
f
y
x
y

Evaluating a Formula

To evaluate the truth value of a formula, we need to replace the variables by concrete objects in the domain. However, we don't necessarily have to perform this substitution for every variable.

There are two types of variables in a formula:

- A variable may be *free*. To evaluate the formula, we need to replace a free variable by an object in the domain.
- A variable may be *bound by a quantifier*. The quantifier tells us how to evaluate the formula.

We need to understand *how to determine whether a variable is free/bound* and *how to replace a free variable with an object in the domain*.

Free and Bound Variables

In a formula $(\forall x\ \alpha)$ or $(\exists x\ \alpha)$, the *scope* of a quantifier is the formula α. A quantifier *binds* its variable within its scope.

An occurrence of a variable in a formula is *bound* if it lies in the scope of some quantifier of the same variable. Otherwise the occurrence of this variable is *free*.

- If a variable occurs multiple times, we need to consider each occurrence of the variable separately.
- The variable symbol immediately after $\exists$ or $\forall$ is neither free nor bound.

A formula with no free variables is called a *closed formula* or *sentence*.

Determine whether a variable is free or bound

Determine whether a variable is free or bound using a parse tree.

1. Draw the parse tree for the formula.
2. Choose the leaf node for an occurrence of a variable.
3. Walk up the tree. Stop as soon as we encounter a quantifier for this variable or we reach the root of the tree.
4. If we *encountered a quantifier for the variable*, this occurrence of the variable is *bound*.
5. If we *reached the root of the tree which is not a quantifier for the variable*, this occurrence of the variable is *free*.

Example 1: $((\forall x\,(P(x) \wedge Q(x))) \to (\neg(P(f(x,y)) \vee Q(y))))$

Example 2: $(\forall x\,((P(x) \wedge Q(x)) \to (\neg(P(f(x,y)) \vee Q(y)))))$

Substitution

When writing natural deduction proofs in predicate logic, it is often useful to replace a variable in a formula with a term.

Suppose that the following sentences are true:

$$(\forall x\,(\mathit{Fish}(x) \to \mathit{Swim}(x)))\,(1)$$

$$\mathit{Fish}(\mathit{Nemo})\,(2)$$

To conclude that Nemo can swim, we need to replace every occurrence of the variable x in the implication $\mathit{Fish}(x) \to \mathit{Swim}(x)$ by the term *Nemo*. This gives us

$$(\mathit{Fish}(\mathit{Nemo}) \to \mathit{Swim}(\mathit{Nemo}))\,(3)$$

By *modus ponens* on (2) and (3), we conclude that $\mathit{Swim}(\mathit{Nemo})$.

Formally, we use *substitution* to refer to the process of replacing x by *Nemo* in the formula $(\forall x\ \mathit{Fish}(x) \to \mathit{Swim}(x))$.

Substitution

Intuitively, $\alpha[t/x]$ answers the question,

> "What happens to α if x has the value specified by term t?"

For a variable x, a term t, and a formula α, $\alpha[t/x]$ denotes the resulting formula by replacing each *free* occurrence of x in α with t. In other words, substitution *does NOT* affect *bound* occurrences of the variable.

Examples of Substitution

Examples.

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

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

Examples: Substitution

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

Examples: Substitution

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

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

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

Examples: Substitution

Example. Let β 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 β 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.

Preventing Capture

Example. Formula $\alpha = (S(x) \wedge (\forall y\, (P(x) \rightarrow Q(y))))$; 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 by a different choice of variable. (Above, we might be able to substitute $f(z, z)$ instead of $f(y, y)$. Or alter α to quantify some other variable.)

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 α,

1. If α is $P(t_1, \dots, t_k)$, then $\alpha[t/x]$ is $P\Big(t_1[t/x], \dots, t_k[t/x]\Big)$.
2. If α is $(\neg\beta)$, then $\alpha[t/x]$ is $(\neg\beta[t/x])$.
3. If α 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 α:

$\vdots$

4. If α is $(\forall x\ \beta)$ or $(\exists x\ \beta)$, then $\alpha[t/x]$ is α.
5. If α is $(\mathcal{Q}y\ \beta)$ for some other variable y and some quantifier $\mathcal{Q} \in \{\exists, \forall\}$, then
   (a) If y does not occur in t, then $\alpha[t/x]$ is $(\mathcal{Q}y\ \beta[t/x])$.
   (b) Otherwise, select a variable z that occurs in neither α nor t; then $\alpha[t/x]$ is $(\mathcal{Q}z\ (\beta[z/y])[t/x])$.

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

Example, Revisited

Example. If α is $\bigl(\forall x\,(\exists y\,((x+y)=z))\bigr)$, 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 α.

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

Example, Revisited

Example. If α 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 α.

Let β be $((x+y)=z)$; thus α 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

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