Predicate Logic: Syntax

Alice Gao

Lecture 12

Based on work by J. Buss, L. Kari, A. Lubiw, B. Bonakdarpour, D. Maftuleac, C. Roberts, R. Trefler, and P. Van Beek

Outline

Learning goals

By the end of this lecture, you should be able to

- Define term.
- Define formula.
- Define free and bound variables.
- Determine whether a variable in a formula is free or bound.
- Determine the scope of a quantifier in a formula.
- Describe the problem when a variable is captured in a substitution.
- Perform substitution in a formula to avoid capture.

The Language of Predicate Logic

- Domain: a non-empty set of objects
- Constants: concrete objects in the domain
- Variables: placeholders for concrete objects in the domain
- Functions: takes objects in the domain as arguments and returns an object of the domain.
- Predicates: takes objects in the domain as arguments and returns true or false. They describe properties of objects or relationships between objects.
- Quantifiers: for how many objects in the domain is the statement true?

Why do we use functions?

Consider two translations of the sentence "every child is younger than its mother."

1. $(\forall \mathrm{x}(\forall \mathrm{y}((\mathrm{Child}(\mathrm{x}) \wedge \mathrm{Mother}(\mathrm{y}, \mathrm{x})) \rightarrow \mathrm{Younger}(\mathrm{x}, \mathrm{y}))))$
2. $(\forall \mathrm{x}(\mathrm{Child}(\mathrm{x}) \rightarrow \mathrm{Younger}(\mathrm{x}, \mathrm{Mother}(\mathrm{x}))))$

Which of the following is the best answer?

(A) Both are wrong.

(B) 1 is correct and 2 is wrong.

(C) 2 is correct and 1 is wrong.

(D) Both are correct. 1 is better.

(E) Both are correct. 2 is better.

The domain is the set of people. $\mathrm{Child}(\mathrm{x})$ means x is a child. $\mathrm{Mother}(\mathrm{x}, \mathrm{y})$ means x is y's mother. $\mathrm{Younger}(\mathrm{x}, \mathrm{y})$ means x is younger than y. $\mathrm{mother}(\mathrm{x})$ returns x's mother.

Why do we use functions?

Using functions allows us to avoid ugly/inelegant predicate logic formulas.

Try translating the following sentence with and without functions.
"Andy and Paul have the same maternal grandmother."

The Language of Predicate Logic

The seven kinds of symbols:

- Constant symbols. Usually lowercase letters
- Variables. Usually lowercase letters
- Function symbols. Usually lowercase letters
- Predicate symbols. Usually uppercase letters
- Connectives: $\neg$, $\wedge$, $\vee$, $\rightarrow$, and $\leftrightarrow$
- Quantifiers: $\forall$ and $\exists$
- Punctuation: '(', ')',and ','

Function symbols and predicate symbols have an assigned arity—the number of arguments required. For example,

- $\mathrm{f}^{(1)}$: f is a unary function.
- $\mathrm{P}^{(2)}$: P is a binary predicate.

Two kinds of expressions

There are two kinds of expressions:

- A term refers to an object in the domain.
- A formula evaluates to T or F.

Predicate Terms

Each term refers to an object of the domain.
We define the set of predicate terms inductively as follows.

1. Each constant symbol is an atomic term.
2. Each variable is an atomic term.
3. $\mathrm{f}(\mathrm{t}_1, \ldots, \mathrm{t}_\mathrm{n})$ is a term if $\mathrm{t}_1, \ldots, \mathrm{t}_\mathrm{n}$ are terms and f is an n-ary function symbol. (If f is a binary function symbol, then we may write $(\mathrm{t}_1\ \mathrm{f}\ \mathrm{t}_2)$ instead of $\mathrm{f}(\mathrm{t}_1, \mathrm{t}_2)$.)
4. Nothing else is a term.

Which expressions are terms?

A term refers to an object of the domain.

Which of the following expressions is a term?

(A) $\mathrm{g}(\mathrm{d}, \mathrm{d})$

(B) $\mathrm{P}(\mathrm{f}(\mathrm{x}, \mathrm{y}), \mathrm{d})$

(C) $\mathrm{f}(\mathrm{x}, \mathrm{g}(\mathrm{y}, \mathrm{z}), \mathrm{d})$

(D) $\mathrm{g}(\mathrm{x}, \mathrm{f}(\mathrm{y}, \mathrm{z}), \mathrm{d})$

Let d be a constant symbol. Let P be a predicate symbol with 2 arguments. Let f be a function symbol with 2 arguments and g be a function symbol with 3 arguments. Let x, y, and z be variable symbols.

Is this a term?

True or False: The expression $(2 - \mathrm{f}(\mathrm{x})) + (\mathrm{y} * \mathrm{x})$ is a term.

(A) True

(B) False

(C) Not enough information to tell

The domain is the set of integers. $+$, $-$ and $*$ are binary functions. f is a unary function. x and y are variables and 2 is a constant.

Predicate Formulas

We define the set of well-formed predicate formulas inductively as follows.

1. $\mathrm{P}(\mathrm{t_1}, \ldots, \mathrm{t_n})$ is an atomic formula if P is an n-ary predicate symbol and each $\mathrm{t_i}$ is a term $(1 \leq \mathrm{i} \leq \mathrm{n})$.
2. $(\neg\alpha)$ is a formula if α is a formula.
3. $(\alpha \star \beta)$ is a formula if α and β are formulas and $\star$ is a binary connective symbol.
4. Each of $(\forall \mathrm{x}\ \alpha)$ and $(\exists \mathrm{x}\ \alpha)$ is a formula if α is a formula and x is a variable.
5. Nothing else is a formula.

Determine whether a formula is well-formed

Constant symbols: m. Variable symbols: x and y. Predicate symbols: $\mathrm{P}^{(2)}$ and $\mathrm{Q}^{(2)}$. Function symbols: $\mathrm{f}^{(1)}$.

Which of the following is a well-formed predicate formula?

(A) $(\mathrm{f}(\mathrm{x}) \to \mathrm{P}(\mathrm{x}, \mathrm{y}))$

(B) $(\forall \mathrm{y}\ \mathrm{P}(\mathrm{m}, \mathrm{f}(\mathrm{y})))$

(C) $(\mathrm{P}(\mathrm{x}, \mathrm{y}) \to \mathrm{Q}(\mathrm{Q}(\mathrm{x})))$

(D) $\mathrm{Q}(\mathrm{m}, \mathrm{f}(\mathrm{m}))$

(E) $\mathrm{P}(\mathrm{m}, \mathrm{f}(\mathrm{Q}(\mathrm{x}, \mathrm{y})))$

Determine whether a formula is well-formed

Things to consider:

- Are there enough brackets? Are the brackets in the right places?
- Is each unary/binary connective applied to the right number of predicates?
- Does every function have the right number and type of arguments?
- Does every predicate have the right number and type of arguments?

Comparing the Definitions of WFFs

Let's compare and contrast the definitions of WFFs for propositional and predicate logic.

- Which parts of the two definitions are the same?
- The definition of WFF for propositional logic says that a propositional variable is an atomic WFF. Is this still the case for predicate logic?
- What is new in the definition of WFF for predicate logic compared to the definition of WFF for propositional logic?
- If we were to prove that every WFF for predicate logic has a property P by structural induction, what does the proof look like? What are the base case(s) and what are the cases in the induction step?

Parse Trees of Predicate Logic Formulas

New elements in the parse tree:

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

Example 1: $(\forall \mathrm{x}(\mathrm{P}(\mathrm{x}) \wedge \mathrm{Q}(\mathrm{x}))) \rightarrow (\neg(\mathrm{P}(\mathrm{f}(\mathrm{x}, \mathrm{y})) \vee \mathrm{Q}(\mathrm{y})))$
Example 2: $(\forall \mathrm{x}((\mathrm{P}(\mathrm{x}) \wedge \mathrm{Q}(\mathrm{x})) \rightarrow (\neg(\mathrm{P}(\mathrm{f}(\mathrm{x}, \mathrm{y})) \vee \mathrm{Q}(\mathrm{y})))))$

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 $(\forall \mathrm{x}\ \alpha)$ or $(\exists \mathrm{x}\ \alpha)$, 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 when 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: $(\forall \mathrm{x}(\mathrm{P}(\mathrm{x}) \wedge \mathrm{Q}(\mathrm{x}))) \rightarrow (\neg(\mathrm{P}(\mathrm{f}(\mathrm{x}, \mathrm{y})) \vee \mathrm{Q}(\mathrm{y})))$
Example 2: $(\forall \mathrm{x}((\mathrm{P}(\mathrm{x}) \wedge \mathrm{Q}(\mathrm{x})) \rightarrow (\neg(\mathrm{P}(\mathrm{f}(\mathrm{x}, \mathrm{y})) \vee \mathrm{Q}(\mathrm{y})))))$

Substitution

Variables are placeholders. We need some means of replacing variables with concrete information.
Suppose that the following sentences are true:

$$(\forall \mathrm{x}(\mathrm{F}(\mathrm{x}) \rightarrow \mathrm{S}(\mathrm{x}))) \, (1)$$

$$\mathrm{F}(\mathrm{Nemo}) \, (2)$$

To conclude that Nemo can swim, we need to replace every occurence of the variable x in the implication $\mathrm{F}(\mathrm{x}) \rightarrow \mathrm{S}(\mathrm{x})$ by the term Nemo. This gives us

$$(\mathrm{F}(\mathrm{Nemo}) \rightarrow \mathrm{S}(\mathrm{Nemo})) \, (3)$$

By modus ponens on (2) and (3), we conclude that $\mathrm{S}(\mathrm{Nemo})$.
Formally, we use substitution to refer to the process of replacing x by Nemo in the formula $(\forall \mathrm{x} \, (\mathrm{F}(\mathrm{x}) \rightarrow \mathrm{S}(\mathrm{x})))$.

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.

Substitution does NOT affect bound occurrences of the variable.

Examples of Substitution

- If α is the formula $\mathrm{E}\,(\mathrm{f}(\mathrm{x}))$, then $\alpha[\mathrm{y}+\mathrm{y}/\mathrm{x}]$ is $\mathrm{E}\,(\mathrm{f}(\mathrm{y}+\mathrm{y}))$.
- $\alpha[\mathrm{f}(\mathrm{x})/\mathrm{x}]$ is $\mathrm{E}\,(\mathrm{f}\,(\mathrm{f}(\mathrm{x})))$.
- $\mathrm{E}\,(\mathrm{f}(\mathrm{x}+\mathrm{y}))\,[\mathrm{y}/\mathrm{x}]$ is $\mathrm{E}\,(\mathrm{f}(\mathrm{y}+\mathrm{y}))$.
- If β is $(\forall \mathrm{x}\ (\mathrm{E}(\mathrm{f}(\mathrm{x})) \wedge \mathrm{S}(\mathrm{x},\mathrm{y})))$ then $\beta[\mathrm{g}(\mathrm{x},\mathrm{y})/\mathrm{x}]$ is β, because β has no free occurrence of x.

Examples: Substitution

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

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

Example. Let β be $(\forall \mathrm{x}\ (\exists \mathrm{y}\ ((\mathrm{x} + \mathrm{y}) = \mathrm{z})))$. What is $\beta[\mathrm{y} - 1/\mathrm{z}]$?

At first thought, we might say $(\forall \mathrm{x}\ (\exists \mathrm{y}\ ((\mathrm{x} + \mathrm{y}) = (\mathrm{y} - 1))))$. But there's a problem—the free variable y in the term $(\mathrm{y} - 1)$ got "captured" by the quantifier $\exists \mathrm{y}$.

We want to avoid this capture.

Preventing Capture

Example.
Let α be $(\mathrm{S}(\mathrm{x}) \wedge (\forall \mathrm{y}\ \ (\mathrm{P}(\mathrm{x}) \rightarrow \mathrm{Q}(\mathrm{y}))))$.
Let t be $\mathrm{f}(\mathrm{y}, \mathrm{y})$.
What is $\alpha[\mathrm{t}/\mathrm{x}]$?

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

We can prevent capture of variables by renaming variables in α.

Substitution—Formal Definition

Let α be a formula, x be a variable, and t be a term.

We compute $\alpha[t/x]$ as follows:

1. If α is $P(t_1, \ldots, t_k)$, then $\alpha[t/x]$ is $P(t_1[t/x], \ldots, t_k[t/x])$
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. If α is $(Qx\ \beta)$, then $\alpha[t/x]$ is α.
5. If α 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 α 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.

Example, Revisited

Example. If α 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 α.
Let β be $(x + y = z)$; thus α is $(\forall x\ (\exists y\ \beta))$.
Select a new variable, say w. Then

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

and

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

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

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

CQ Substitution

Let α be the predicate formula below.

$$((\forall \mathrm{y}\ ((\forall \mathrm{x}\ \mathrm{P}(\mathrm{x})) \wedge \mathrm{P}(\mathrm{y}))) \rightarrow (\forall \mathrm{y}\ \mathrm{Q}(\mathrm{x}, \mathrm{y}))).$$

Consider the four leaf nodes of the parse tree of the above formula. How many of the four occurrences of the variables are bound?

(A) 0

(B) 1

(C) 2

(D) 3

(E) 4

CQ Substitution

Let α be the predicate formula below.

$$((\forall \mathrm{y}\ ((\forall \mathrm{x}\ \mathrm{P}(\mathrm{x})) \wedge \mathrm{P}(\mathrm{y}))) \rightarrow (\forall \mathrm{y}\ \mathrm{Q}(\mathrm{x}, \mathrm{y}))).$$

If we perform the substitution $\alpha[\mathrm{y}/\mathrm{x}]$, does capture occur?

(A) Yes.

(B) No.

(C) I don't know.

CQ Substitution

Let α be the predicate formula below.

$$((\forall \mathrm{y}\ ((\forall \mathrm{x}\ \mathrm{P}(\mathrm{x})) \wedge \mathrm{P}(\mathrm{y}))) \rightarrow (\forall \mathrm{y}\ \mathrm{Q}(\mathrm{x}, \mathrm{y}))).$$

Following our substitution algorithm to prevent capture, which of the two occurrences of y are renamed?

(A) Both are renamed.

(B) From the left, $\mathrm{y}(1)$ is renamed and $\mathrm{y}(2)$ is NOT renamed.

(C) From the left, $\mathrm{y}(1)$ is NOT renamed and $\mathrm{y}(2)$ is renamed.

(D) Neither is renamed.

CQ Substitution

Let α be the predicate formula below.

$$((\forall \mathrm{y}\ ((\forall \mathrm{x}\ \mathrm{P}(\mathrm{x})) \wedge \mathrm{P}(\mathrm{y}))) \rightarrow (\forall \mathrm{y}\ \mathrm{Q}(\mathrm{x}, \mathrm{y}))).$$

Following our substitution algorithm to prevent capture, which of the two occurrences of x are replaced by y?

(A) Both are replaced.

(B) From the left, $\mathrm{x}(1)$ is replaced and $\mathrm{x}(2)$ is NOT replaced.

(C) From the left, $\mathrm{x}(1)$ is NOT replaced and $\mathrm{x}(2)$ is replaced.

(D) Neither is replaced.

Revisiting the learning goals

By the end of this lecture, you should be able to

- Define term.
- Define formula.
- Define free and bound variables.
- Determine whether a variable in a formula is free or bound.
- Determine the scope of a quantifier in a formula.
- Describe the problem when a variable is captured in a substitution.
- Perform substitution in a formula to avoid capture.