

華東師範大學
EAST CHINA NORMAL
UNIVERSITY

Logic in Computer Science

Lecture 03

Predicate Logic

Li Qin

Associate Professor

School of Software Engineering

- ★ The need for a richer language
- ★ Predicate logic as a formal language
 - terms —variables, functions
 - formulas —predicates, quantifiers
 - free and bound variables
 - substitution
- ★ Proof theory of predicate logic
 - Natural deduction rules

Propositional Logic:

- Study of declarative sentences, statements about the world which can be given a truth value
- Dealt very well with sentence components like: *not*, *and*, *or*, *if ... then ...*
- **Limitations:** cannot deal with modifiers like *there exists*, *all*, *among*, *only*.

Example: “*Every student is younger than some instructor.*”

- We could identify the entire phrase with the propositional symbol *p*.
- However, the phrase has a finer logical structure. It is a statement about the following properties:
 - being a student
 - being an instructor
 - being younger than somebody else

Properties are expressed by predicates. S , I , Y are *predicates*.

$S(\text{andy})$: Andy is a *student*.

$I(\text{paul})$: Paul is an *instructor*.

$Y(\text{andy}, \text{paul})$: Andy is *younger than* Paul.

Variables are placeholders for concrete values.

$S(x)$: x is a student.

$I(x)$: x is an instructor.

$Y(x, y)$: x is younger than y .

Quantifiers make possible encoding the phrase:

“Every student is younger than some instructor.”

Two quantifiers: \forall —*forall*, and \exists —*exists*.

Encoding of the above sentence:

$$\forall x(S(x) \rightarrow (\exists y(I(y) \wedge Y(x, y))))$$

“No books are gaseous. Dictionaries are books. Therefore, no dictionary is gaseous.”

We denote: $B(x)$: x is a book

$G(x)$: x is gaseous

$D(x)$: x is a dictionary

$$\neg \exists x (B(x) \wedge G(x)), \forall x (D(x) \rightarrow B(x))$$

\vdash

$$\neg \exists x (D(x) \wedge G(x))$$

“Every child is younger than his mother”

We denote: $C(x)$: x is a child

$M(x, y)$: x 's mother is y

$$\forall x \forall y (C(x) \wedge M(x, y) \rightarrow Y(x, y))$$

Denote $m(x)$: mother of x

$$\forall x (C(x) \rightarrow Y(x, m(x)))$$

Using the function m to encode the “mother of” relationship is more appropriate, since every person has a unique mother.

“Andy and Paul have the same maternal grandmother”

$$\forall x \forall y \forall u \forall v (M(a, x) \wedge M(x, y) \wedge M(p, u) \wedge M(u, v) \rightarrow y = v)$$

We have introduced a new, special predicate: *equality*.

Alternative representation:

$$m(m(a)) = m(m(p))$$

Consider the relationship $B(x, y)$: x is the brother of y . This relationship must be encoded as a predicate, since a person may have more than one brother.

Two sorts of “things” in a predicate formula:

- Objects such as *a* (Andy) and *p* (Paul). Function symbols also refer to objects. These are modeled by *terms*.
- Expressions that can be given truth values. These are modeled by *formulas*.

A predicate vocabulary consists of 3 sets:

- *Predicate symbols* \mathcal{P} ;
 - *Function symbols* \mathcal{F} ;
 - *Constants* \mathcal{C} .
- } Each predicate and function symbol comes with a fixed *arity* (i.e. number of arguments)

Elements of the formal language of predicate logic:

- Terms
- Formulas
- Free and bound variables
- Substitution

Definition: *Terms* are defined as follows:

- Any variable is a term;
- Any constant in \mathcal{C} is a term;
- If t_1, \dots, t_n are terms and $f \in \mathcal{F}$ has arity n , then $f(t_1, \dots, t_n)$ is a term;
- Nothing else is a term.

Backus-Naur definition: $t ::= x \mid c \mid f(t, \dots, t)$ where x represents variables, c represents constants in \mathcal{C} , and f represents function in \mathcal{F} with arity n .

Remarks:

- The first building blocks of terms are constants and variables.
- More complex terms are built from function symbols using previously built terms.
- The notion of terms is independent on the sets \mathcal{C} and \mathcal{F} .

Definition: We define the set of *formulas* over $(\mathcal{F}, \mathcal{P})$ inductively, using the already defined set of terms over \mathcal{F} .

- If P is a predicate with $n \geq 1$ arguments, and t_1, \dots, t_n are terms over \mathcal{F} , then $P(t_1, \dots, t_n)$ is a formula.
- If Φ is a formula, then so is $\neg\Phi$.
- If Φ and Ψ are formulas, then so are $\Phi \wedge \Psi$, $\Phi \vee \Psi$, $\Phi \rightarrow \Psi$.
- If Φ is a formula and x is a variable, then $\forall x\Phi$ and $\exists x\Phi$ are formulas.
- Nothing else is a formula.

BNF definition:

$$\Phi ::= P(t_1, \dots, t_n) \mid (\neg\Phi) \mid (\Phi \wedge \Phi) \mid (\Phi \vee \Phi) \mid (\Phi \rightarrow \Phi) \mid (\forall x\Phi) \mid (\exists x\Phi)$$

where P is a predicate of arity n , t_i are terms, $i \in \{1, \dots, n\}$, x is a variable.

Convention: We retain the usual binding priorities of the connectives $\neg, \wedge, \vee, \rightarrow$. We add that $\forall x$ and $\exists x$ bind like \neg .

Example

Consider translating the sentence:

“Every son of my father is my brother”

Two alternatives:

- “*Father of*” relationship encoded as a predicate.

$S(x, y)$: x is the son of y .

$F(x, y)$: x is the father of y .

$B(x, y)$: x is the brother of y .

m : constant, denoting “myself”.

Translation: $\forall x \forall y (F(x, m) \wedge S(y, x) \rightarrow B(y, m))$

- “*Father of*” relationship encoded as a function.

$f(x)$: father of x .

Translation: $\forall x (S(x, f(m)) \rightarrow B(x, m))$

Definition: Let Φ be a formula in predicate logic. An occurrence of x in Φ is *free in Φ* if it is a leaf node in the parse tree of Φ such that there is no path upwards from that node x to a node $\forall x$ or $\exists x$. Otherwise, that occurrence x is called *bound*. For $\forall x\Phi$, we say that Φ —minus any of its sub-formulas $\exists x\Psi$, or $\forall x\Psi$ —is the scope of $\forall x$, respectively $\exists x$.

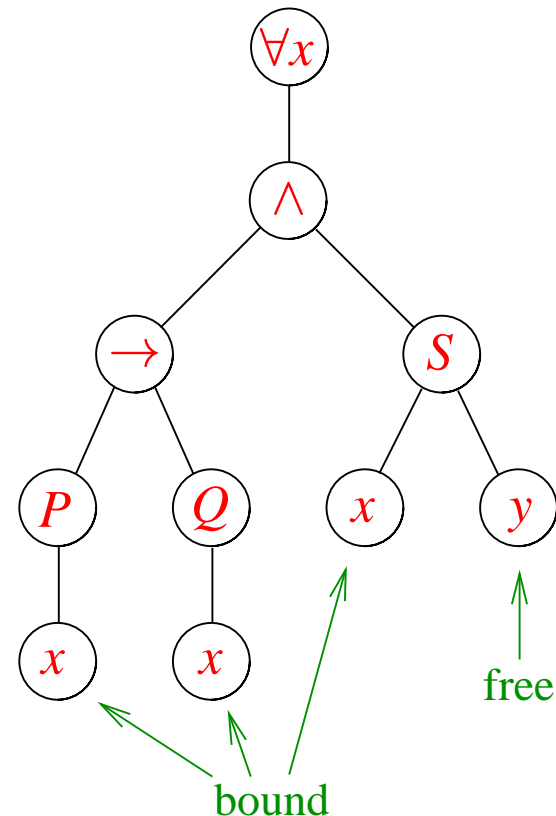
Formula:

$$\forall x ((P(x) \rightarrow Q(x)) \wedge S(x, y))$$

Scope of $\forall x$

x is bound.

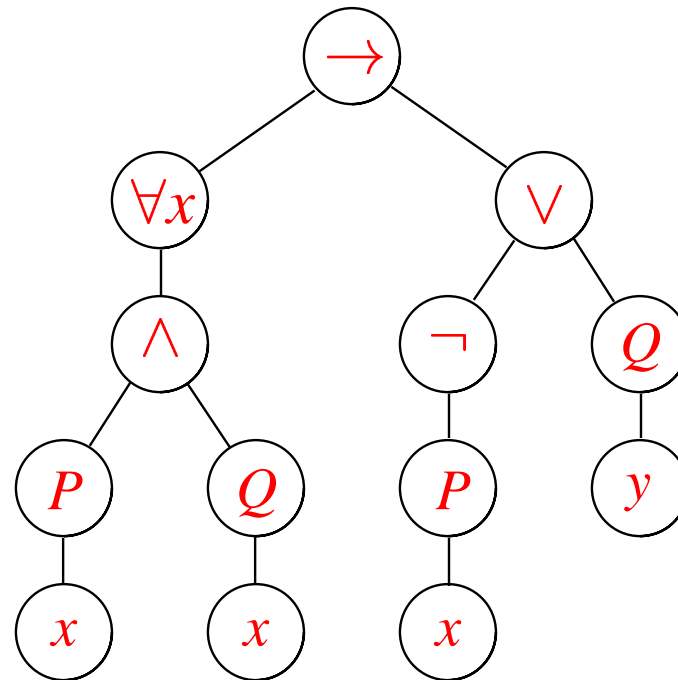
y is free.



Examples of Free and Bound Variables

Formula: $(\forall x(P(x) \wedge Q(x))) \rightarrow (\neg P(x) \vee Q(y))$

Parse tree:

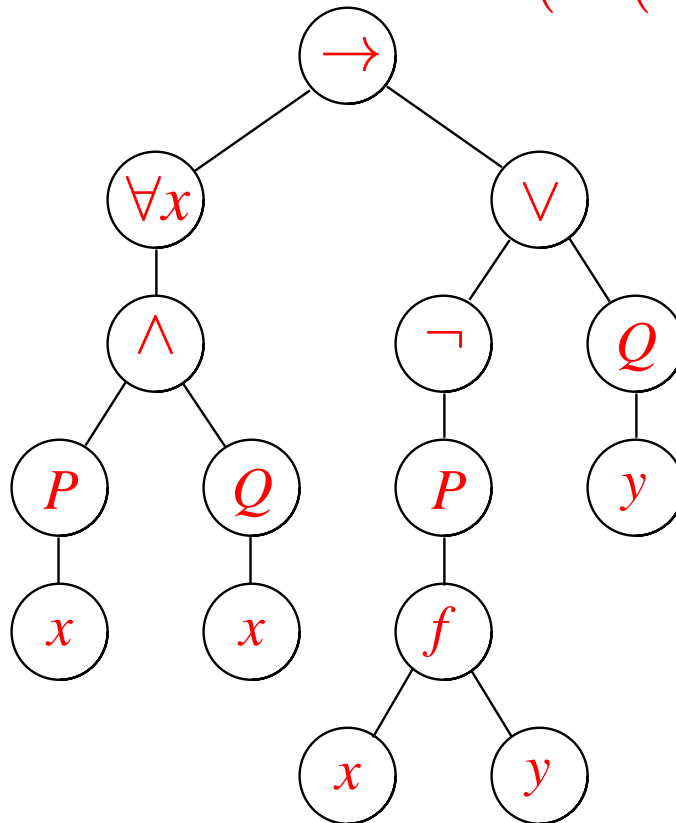


Substitution

Variables are placeholders, so we must have means of *replacing* them with more concrete information.

Definition: Given a variable x , a term t , and a formula Φ , we define $\Phi[t/x]$ to be the formula obtained by replacing each free occurrence of variable x in Φ with t .

$((\forall x (P(x) \wedge Q(x))) \rightarrow (\neg P(x) \vee Q(y)))[f(x,y)/x]$ is
 $(\forall x (P(x) \wedge Q(x))) \rightarrow (\neg P(f(x,y)) \vee Q(y))$



Substitution (2)

Definition: Given a term t , a variable x , and a formula Φ , we say that t *is free for x in Φ* if no free x leaf in Φ occurs in the scope of $\forall y$ or $\exists y$, for every variable y occurring in t .

Remark: If t is not free for x in Φ , then the substitution $\Phi[t/x]$ has unwanted effects.

Example:

$(S(x) \wedge (\forall y (P(x) \rightarrow Q(y))))[y/x]$ is $S(y) \wedge (\forall y (P(y) \rightarrow Q(y)))$

Avoid this by renaming $\forall y$ into $\forall z$.

$(S(x) \wedge (\forall z (P(x) \rightarrow Q(z))))[y/x]$ is $S(y) \wedge (\forall z (P(y) \rightarrow Q(z)))$

- Natural deduction rules for propositional logic are still valid
- Natural deduction rules for predicate logic:
 - proof rules from propositional logic;
 - proof rules for equality;
 - proof rules for universal quantification;
 - proof rules for existential quantification.
- Quantifier equivalences

$$\frac{}{t = t} = \text{i} \qquad \frac{t_1 = t_2 \quad \Phi[t_1/x]}{\Phi[t_2/x]} = \text{e}$$

Convention: When we write a substitution in the form $\Phi[t/x]$, we implicitly assume that t is free for x in Φ .

Proof example:

$$x + 1 = 1 + x, (x + 1 > 1) \rightarrow (x + 1 > 0) \vdash (1 + x > 1) \rightarrow (1 + x > 0)$$

1	$x + 1 = 1 + x$	premise
2	$(x + 1 > 1) \rightarrow (x + 1 > 0)$	premise
3	$(1 + x > 1) \rightarrow (1 + x > 0)$	=e 1,2

Proof Rules for Universal Quantification

$$\frac{\forall x \Phi}{\Phi[t/x]} \quad \forall x e$$

$$\boxed{\begin{array}{c} x_0 \\ \vdots \\ \Phi[x_0/x] \end{array}} \quad \forall x i$$

Proof examples:

$$\forall x (P(x) \rightarrow Q(x)), \forall x P(x) \vdash \forall x Q(x)$$

$$P(t), \forall x (P(x) \rightarrow \neg Q(x)) \vdash \neg Q(t)$$

1	$\forall x (P(x) \rightarrow Q(x))$	premise
2	$\forall x P(x)$	premise
3	$x_0 \quad P(x_0) \rightarrow Q(x_0)$	$\forall x e 1$
4	$P(x_0)$	$\forall x e 2$
5	$Q(x_0)$	$\rightarrow e 3,4$
6	$\forall x Q(x)$	$\forall x i 3-5$

1	$P(t)$	premise
2	$\forall x (P(x) \rightarrow \neg Q(x))$	premise
3	$p(t) \rightarrow \neg Q(t)$	$\forall x e 2$
4	$\neg Q(t)$	$\rightarrow e 3,1$

Proof Rules for Existential Quantification

$$\frac{\Phi[t/x]}{\exists x \Phi} \quad \exists x i$$

$$\frac{\exists x \Phi \quad \boxed{\begin{array}{c} x_0 \quad \Phi[x_0/x] \\ \vdots \\ \chi \end{array}}}{\chi} \quad \text{Side condition: } x_0 \text{ doesn't occur in } \chi \quad \exists x e$$

Proof examples:

$$\forall x (P(x) \rightarrow Q(x)), \exists x P(x) \vdash \exists x Q(x)$$

$$\forall x \Phi \vdash \exists x \Phi$$

1	$\forall x \Phi$	premise
2	$\Phi[x/x]$	$\forall x e$ 1
3	$\exists x \Phi$	$\exists x i$ 2

1	$\forall x (P(x) \rightarrow Q(x))$	premise
2	$\exists P(x)$	premise
3	$\boxed{\begin{array}{c} x_0 \quad P(x_0) \\ \\ P(x_0) \rightarrow Q(x_0) \\ \\ Q(x_0) \\ \\ \exists x Q(x) \end{array}}$	assumption
4		$\forall x e$ 1
5		$\rightarrow e$ 4,3
6		$\exists x i$ 5
7	$\exists x Q(x)$	$\exists x e$ 2,3–6

Another Example

$$\exists x P(x), \forall x \forall y (P(x) \rightarrow Q(y)) \vdash \forall y Q(y)$$

1	$\exists x P(x)$	premise
2	$\forall x \forall y (P(x) \rightarrow Q(y))$	premise
3	y_0	
4	x_0 $P(x_0)$	assumption
5	$\forall y (P(x_0) \rightarrow Q(y))$	$\forall x$ e 2
6	$P(x_0) \rightarrow Q(y_0)$	$\forall y$ e 2
7	$Q(y_0)$	\rightarrow 6,4
8	$Q(y_0)$	$\exists x$ e 1,4-7
9	$\forall y Q(y)$	$\forall y$ i 3–8