

COL703: Logic for Computer Science (Aug-Nov 2022)

Lectures 13 & 14 (Predicate Logic)

Kumar Madhukar

`madhukar@cse.iitd.ac.in`

September 22nd and 29th, 2022

The need for a richer language

- the logical aspects of natural and artificial languages are much richer
 - than what propositional logic can capture
 - limited to sentence components like **not**, **and**, **or**, **if ... then**

- consider the following declarative sentence

Every student is younger than some instructor.

- a propositional atom denoting this fails to capture the finer logical structure of this sentence.

The need for a richer language

Every student is younger than some instructor.

this is about **being a student**, **being an instructor**, and **being younger than somebody else**

we would like a mechanism to express these with their logical relationships

this is what we will use **predicates** for

Predicates

- we can use predicates S , I , and Y
- $S(\text{John})$ – John is a student
- $I(\text{Paul})$ – Paul is an instructor
- $Y(\text{John}, \text{Paul})$ – John is younger than Paul
- the meaning of these symbols must be specified exactly

Predicates

- can we now express "Every student is younger than some instructor."

Predicates

- can we now express "Every student is younger than some instructor."
- we don't want to write down every instance of $S(.)$
- use **variables** as place holders for concrete values
- $S(x)$ – x is a student
- $I(x)$ – x is an instructor
- $Y(x, y)$ – x is younger than y

Quantifiers

- can we now express "Every student is younger than some instructor."

Quantifiers

- can we now express "Every student is younger than some instructor."
- Every student is younger than some instructor.
- \forall (for all) and \exists (there exists)

Quantifiers

- can we now express "Every student is younger than some instructor."
- Every student is younger than some instructor.
- \forall (for all) and \exists (there exists)
- the quantifiers always come attached to a variable
- $\forall x$ (for all x) and $\exists z$ (there exists z)

Example

- can we now express "Every student is younger than some instructor."

Example

- can we now express "Every student is younger than some instructor."
- $\forall x (S(x) \rightarrow (\exists y (I(y) \wedge Y(x, y))))$
- for every x , if x is a student, then there is some y such that y is an instructor and x is younger than y
- predicates can have any (finite) number of arguments (arity)

Another example

- Not all birds can fly.
- $B(x)$ – x is a bird
- $F(x)$ – x can fly

Another example

- Not all birds can fly.
- $B(x)$ – x is a bird
- $F(x)$ – x can fly
- $\neg(\forall x (B(x) \rightarrow F(x)))$

Another example

- Not all birds can fly.
- $B(x)$ – x is a bird
- $F(x)$ – x can fly
- $\neg(\forall x (B(x) \rightarrow F(x)))$
- alternatively, $\exists x (B(x) \wedge \neg F(x))$

Another example

- Not all birds can fly.
- $B(x)$ – x is a bird
- $F(x)$ – x can fly
- $\neg(\forall x (B(x) \rightarrow F(x)))$
- alternatively, $\exists x (B(x) \wedge \neg F(x))$
- does this formula evaluate to true in the world we currently live in?

Another example

Every child is younger than its mother.

Another example

Every child is younger than its mother.

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

Another example

Andy and Paul have the same maternal grandmother.

Another example

Andy and Paul have the same maternal grandmother.

$$\forall x \forall y \forall u \forall v ((M(x, y) \wedge M(y, \textit{Andy}) \wedge M(u, v) \wedge M(v, \textit{Paul})) \rightarrow x = u)$$

Another example

Andy and Paul have the same maternal grandmother.

$$\forall x \forall y \forall u \forall v ((M(x, y) \wedge M(y, \text{Andy}) \wedge M(u, v) \wedge M(v, \text{Paul})) \rightarrow x = u)$$

- function symbols can help us avoid the inelegant encoding
- equality has been used as a special predicate

As a formal language

there are two sorts of things involved in a predicate logic formula:

objects that we are talking about – constants, variables, $m(a)$, $g(x, y)$

expressions denoting objects are called terms

the other sort of things are formulas

Terms

- Any variable is a term.
- If $c \in \mathcal{F}$ is a nullary function, then c is a term.
- If t_1, t_2, \dots, t_n are terms and $f \in \mathcal{F}$ has arity $n > 0$, then $f(t_1, t_2, \dots, t_n)$ is a term.
- Nothing else is a term.

In Backus Naur form we may write

$$t ::= x \mid c \mid f(t, \dots, t)$$

where x ranges over a set of variables **var**, c over nullary function symbols in \mathcal{F} , and f over those elements of \mathcal{F} with arity $n > 0$.

Formulas

- If $P \in \mathcal{P}$ is a predicate symbol of arity $n \geq 1$, and if t_1, t_2, \dots, t_n are terms over \mathcal{F} , then $P(t_1, t_2, \dots, t_n)$ is a formula.
- If ϕ is a formula, then so is $(\neg\phi)$.
- If ϕ and ψ are formulas, then so are $(\phi \wedge \psi)$, $(\phi \vee \psi)$ and $(\phi \rightarrow \psi)$.
- 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.

Note how the arguments given to predicates are always terms. This can also be seen in the Backus Naur form (BNF) for predicate logic:

$$\phi ::= P(t_1, t_2, \dots, t_n) \mid (\neg\phi) \mid (\phi \wedge \phi) \mid (\phi \vee \phi) \mid (\phi \rightarrow \phi) \mid (\forall x \phi) \mid (\exists x \phi)$$

Binding properties

- \neg , $\forall y$, $\exists y$ bind most tightly
- then \vee and \wedge
- then \rightarrow , which is right-associative

Another example

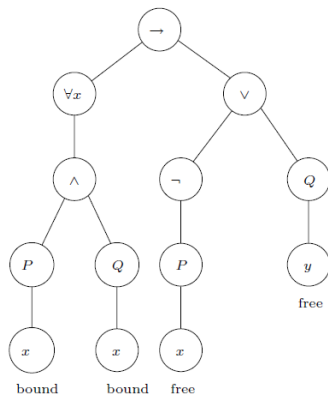
Every son of my father is my brother.

Parse trees

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

Free and bound variables

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



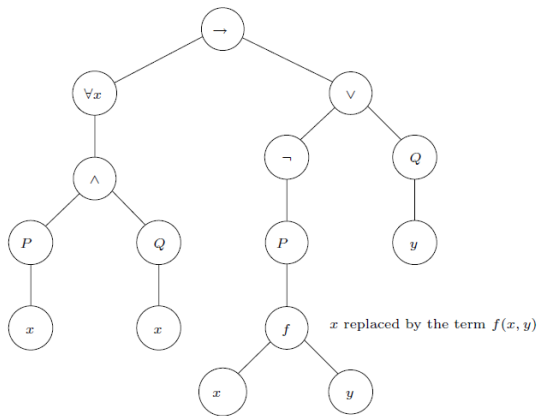
Substitution

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

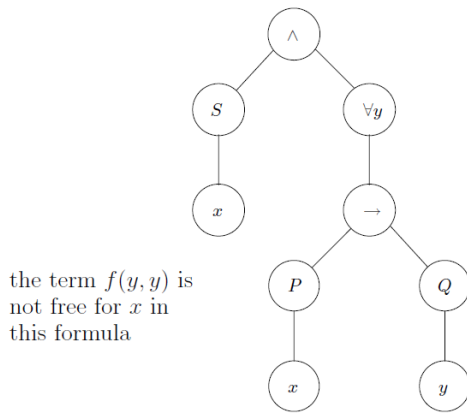
Substitution: example

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



Undesired side-effects of substitution

substitution must be **avoided** if t is not free for x in ϕ



Proof theory (natural deduction rules)

(from the book by Huth and Ryan, pages 107-117)

Quantifier Equivalence

(from the book by Huth and Ryan, pages 117-122)

Next week

- Semantics of FOL
- Normal forms, Herbrand theorem, Ground resolution
- Undecidability result

Thank you!