

Predicate Logic

(aka. First-Order Logic)

Syntax, Semantics, Models

Bettina Könighofer

IAIK – Graz University of Technology
bettina.koenighofer@iaik.tugraz.at

Motivation



- Limits of Propositional Logic
 - Need for more expressive language
- Syntax
 - Construct formulas
- Semantics, Models
 - Understand formulas

Outline

- Intuitive Example
- Syntax
- Models
- Semantics



Intuitive Example



Rex

$$\begin{array}{c}
 \frac{\text{Dog(Rex)}}{p} \\
 \hline
 \text{Rex is a dog.} \\
 \hline
 \frac{\text{Rex has fur.}}{r} \\
 \hline
 \text{Fur(Rex)}
 \end{array}
 \qquad
 \begin{array}{c}
 \frac{\forall x (\text{Dog}(x) \rightarrow \text{Fur}(x))}{q} \\
 \hline
 \text{All dogs have fur.} \\
 \hline
 \text{Fur(Rex)}
 \end{array}$$

$$\circ) p, q \vdash r \qquad p=T, q=T, r=\perp \quad \Downarrow$$

$$\circ) \text{Dog(Rex), } \forall x (\text{Dog}(x) \rightarrow \text{Fur}(x)) \vdash \text{Fur(Rex)} \quad \checkmark$$

More Examples

- Not all birds can fly.

$$\neg \forall x (Bird(x) \rightarrow Fly(x))$$

- Every student is younger than some teacher.

$$\forall x \exists y (Student(x) \wedge Teacher(y) \rightarrow Younger(x, y))$$

- Andi and Paul have the same maternal grandmother.

$$mother(mother(Andi)) = mother(mother(Paul))$$

More Examples

- Some people in class visited the Grand Canyon.

$$\exists x (\text{Person}(x) \wedge \text{InClass}(x) \wedge \text{VisitedGC}(x))$$

- All integers are either even or odd.

$$\forall x (\text{Integer}(x) \rightarrow (\text{Even}(x) \oplus \text{Odd}(x)))$$

More Examples

- „Alice has no sister.“
 - Alice: constant
 - $Sister(x,y)$... x is y 's sister
 - $\neg \exists x \textit{ Sister (Alice, } x)$

- „A person who wears a crown is either a king or a queen.“
 - $P(x)$... x is a person
 - $C(x)$... x wears a crown
 - $K(x)$... x is a king
 - $Q(x)$... x is a queen
 - $\forall x((P(x) \wedge C(x)) \rightarrow (K(x) \oplus Q(x)))$

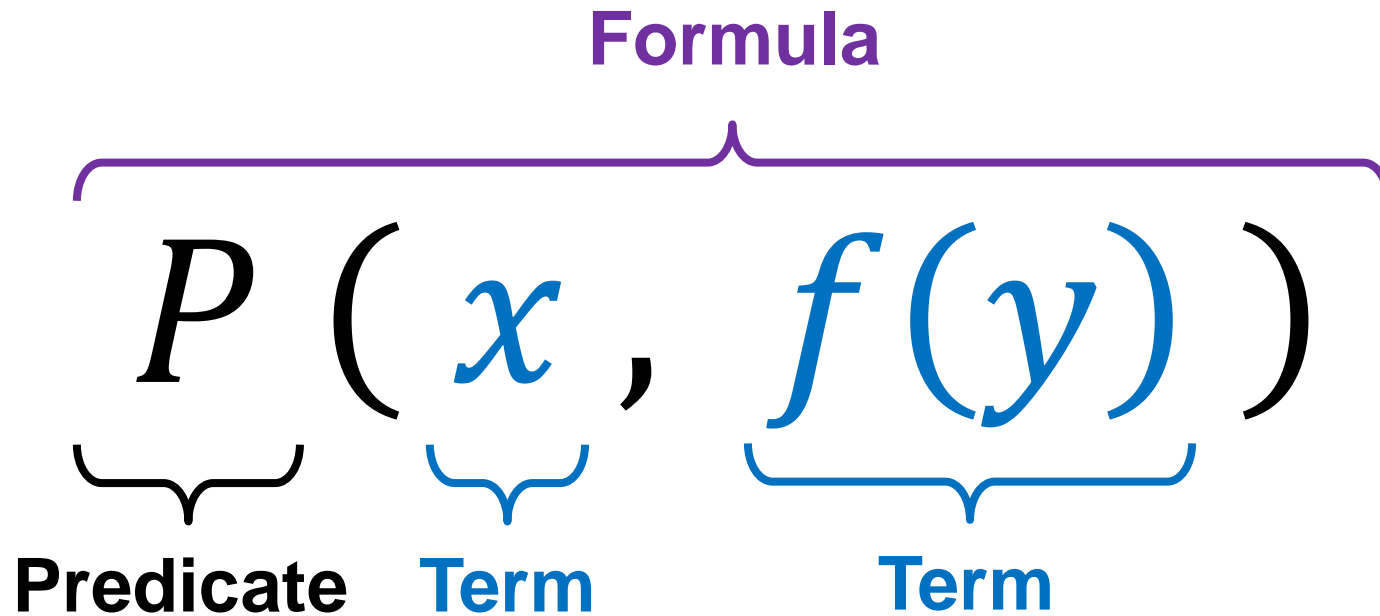
Syntax of Predicate Logic

- Subjects (Objects in OOP)
 - Rex
 - Andy
 - Paul
 - ...
- Truth values
 - Dog (Rex)
 - ...

Terms

Formulas

From Terms to Formulas



Syntax of Predicate Logic

- Variables \mathbb{V}
 - x, y, z, \dots
- Functions \mathbb{F}
 - f, g, h, \dots (arity > 0)
 - constants (arity $= 0$)
- Terms \mathbb{T}
 - defined recursively
- Predicates \mathbb{P}
 - P, Q, R, \dots (arity > 0)
 - Prop. constants (arity $= 0$)

Terms \mathbb{T}

- Variable
- Nullary Function (constant)
- Terms t_1, t_2, \dots, t_n , n -ary Function f
 \Downarrow
 $f(t_1, t_2, \dots, t_n)$ is a term

Formulas

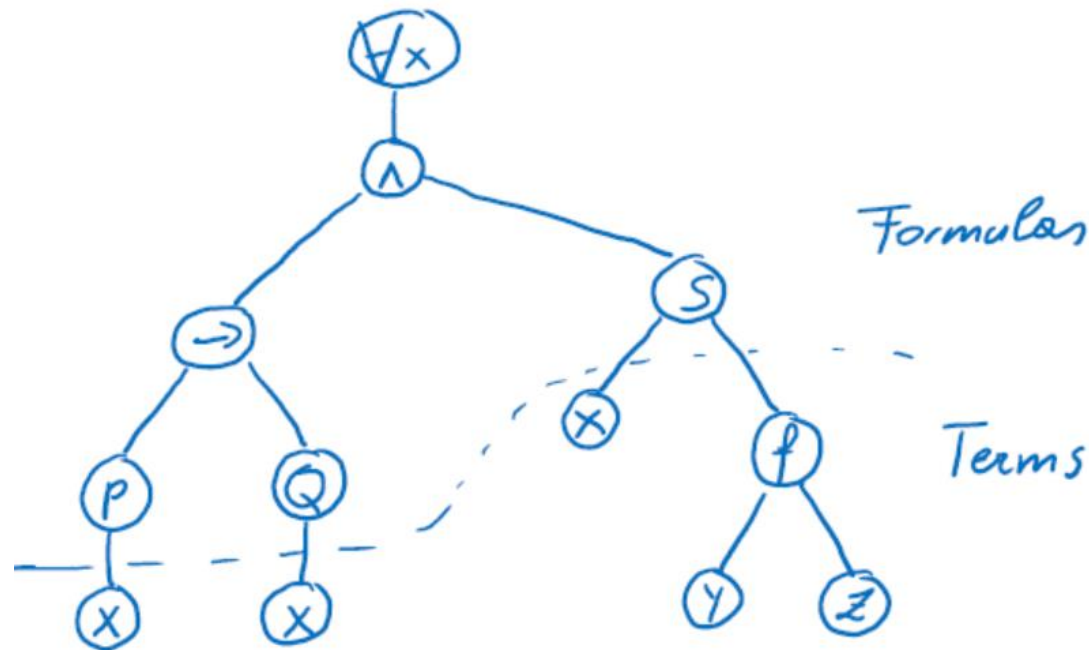
Preconditions:

- Terms t_1, t_2, \dots, t_n
 - n -ary predicate symbol P
 - formulas ϕ, ψ
 - Variable x
- $P(t_1, t_2, \dots, t_n)$
 - $\neg \phi$
 - $\phi \wedge \psi, \phi \vee \psi, \phi \rightarrow \psi$
 - $\forall x \phi$
 - $\exists x \phi$

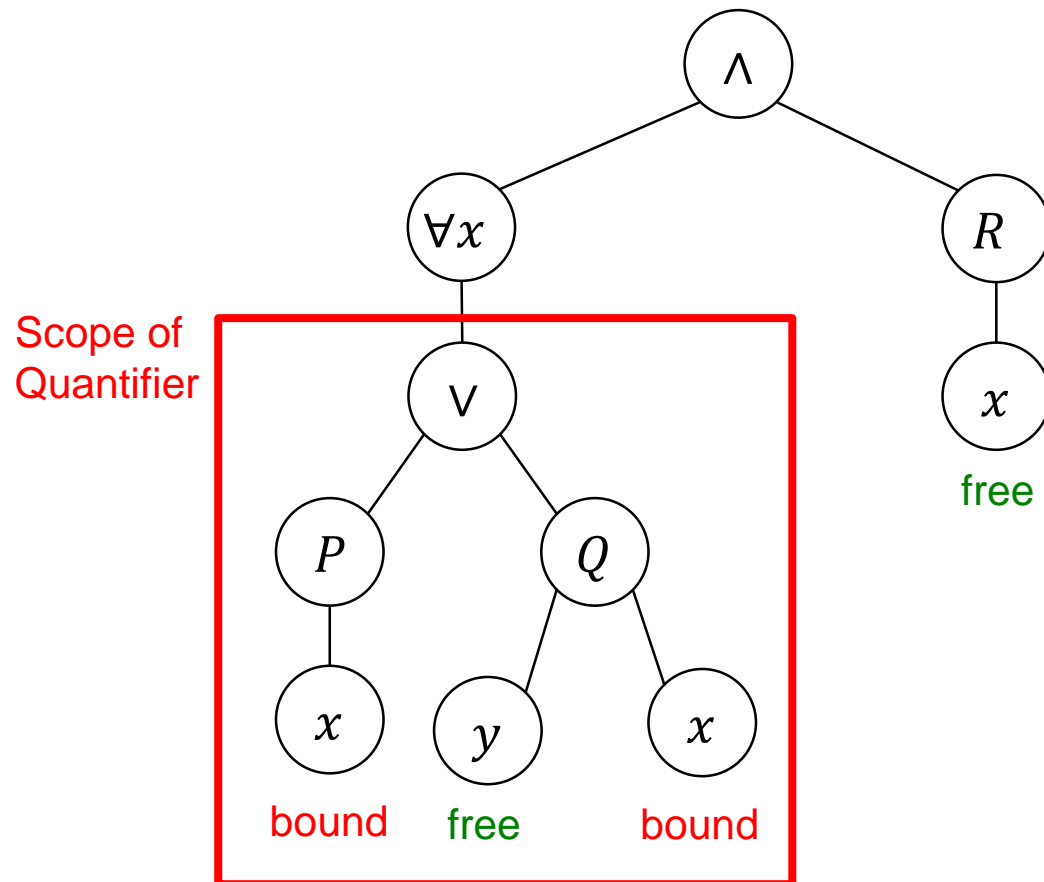
Binding Priorities

1. \forall, \exists, \neg e.g., $\forall x \left((P(x) \rightarrow Q(x)) \wedge S(x, f(y, z)) \right)$
2. \wedge
3. \vee
4. \rightarrow
 - right-associative

Syntax Tree $\forall x \left((P(x) \rightarrow Q(x)) \wedge S(x, f(y, z)) \right)$



Free and Bound Variables



Substitution

$$\phi [a / x]$$

Term

Variable

Read: “ ϕ with a for x ”

Example:

$$\phi = P(f(x, y)) \vee Q(x)$$

$$\phi[a/x] = P(f(a, y)) \vee Q(a)$$

Substitution

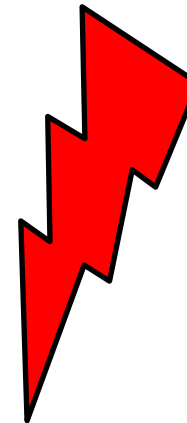
- Only *free* variables
 - $\phi = \forall x (P(x) \wedge Q(x))$
 - $\phi[a/x] = \forall x (P(x) \wedge Q(x)) = \phi$

Side-Condition for Substitution

- Example:

- $\phi = \exists x (P(x) \rightarrow Q(z))$

- $\phi[f(x)/z] = \exists x (P(x) \rightarrow Q(f(x)))$



- No “capturing”:

- Term t free for variable v in formula ϕ

- $f(x)$ **not free for** z in ϕ

Model M for (\mathbb{F}, \mathbb{P})

- Non-empty set A
 - “**Universe of Values**” or “**Domain of Values**”
 - Possibly infinite
- For nullary $f \in \mathbb{F}$: concrete element $f^M \in A$
- For nullary $P \in \mathbb{P}$: **true** or **false**
- For other $f \in \mathbb{F}$: concrete function $f^M: A^n \rightarrow A$
 - Defined by e.g. function table
- For $P \in \mathbb{P}$ with arity $n > 0$: subset $P^M \subseteq A^n$
 - Tuples which make P true
- Look-Up Table for Free Variables
 - Function $l: \mathbb{V} \rightarrow A$

Semantics of Predicate Logic

- Formula ϕ
 - Over \mathbb{F}, \mathbb{P}
- Model M
 - Domain A , For \mathbb{F}, \mathbb{P} , Lookup Table l for \mathbb{V}
- $M \models \phi$?

Semantics of Predicate Logic

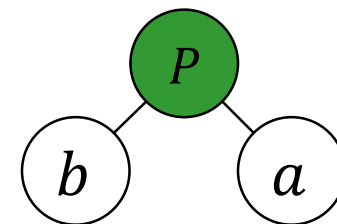
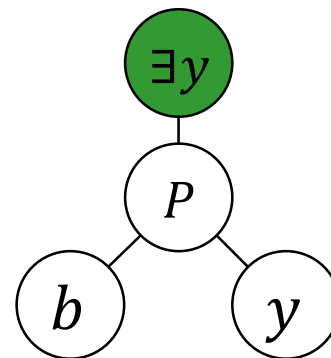
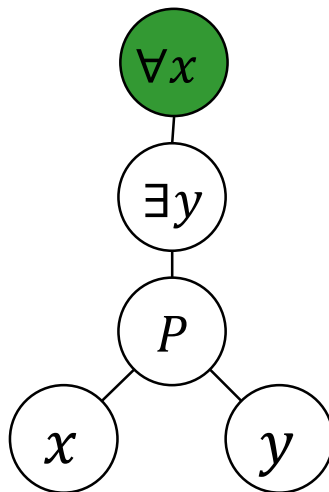
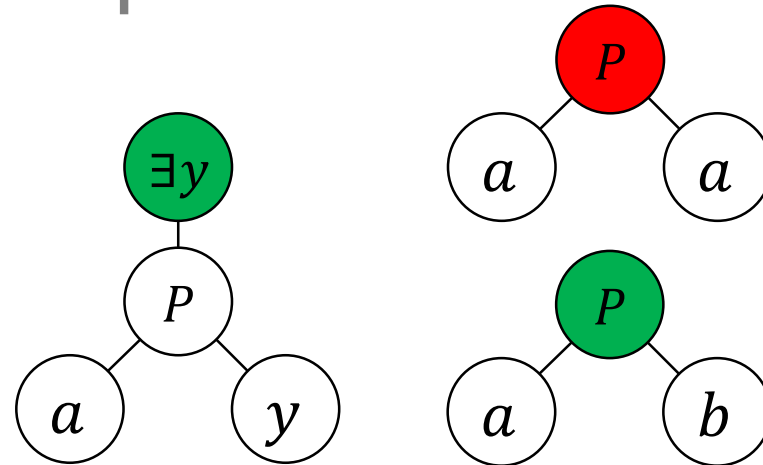
- For ϕ of the form $P(t_1, t_2, \dots, t_n)$
 - Interpret t_1, \dots, t_n via l, M
 - Obtain (a_1, a_2, \dots, a_n)
 - $M \models P(t_1, t_2, \dots, t_n)$ iff $(a_1, a_2, \dots, a_n) \in P^M$

Semantics of Predicate Logic

- For ϕ of the form $\forall x \psi$
 - $M \models \forall x \psi$ iff $M \models \psi$, for all $a \in A$
- For ϕ of the form $\exists x \psi$
 - $M \models \exists x \psi$ iff $M \models \psi$, for at least one $a \in A$
- For ϕ of the form $\neg\psi$, $\psi_1 \wedge \psi_2$, $\psi_1 \vee \psi_2$, $\psi_1 \rightarrow \psi_2$
 - Like in propositional logic

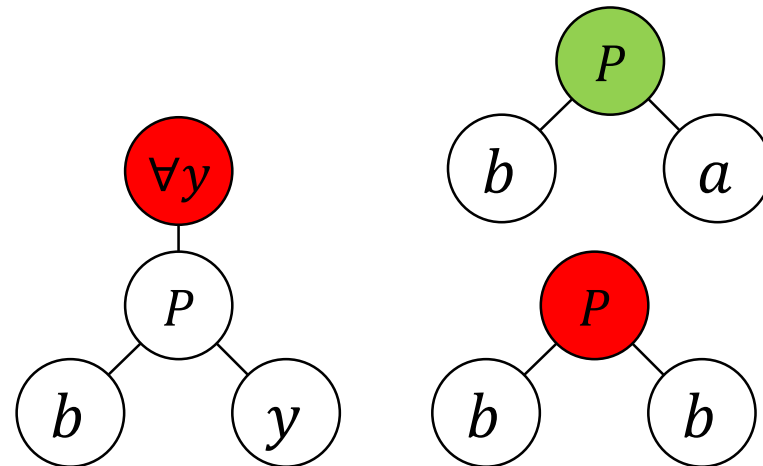
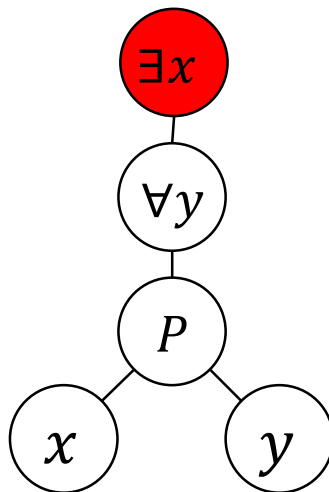
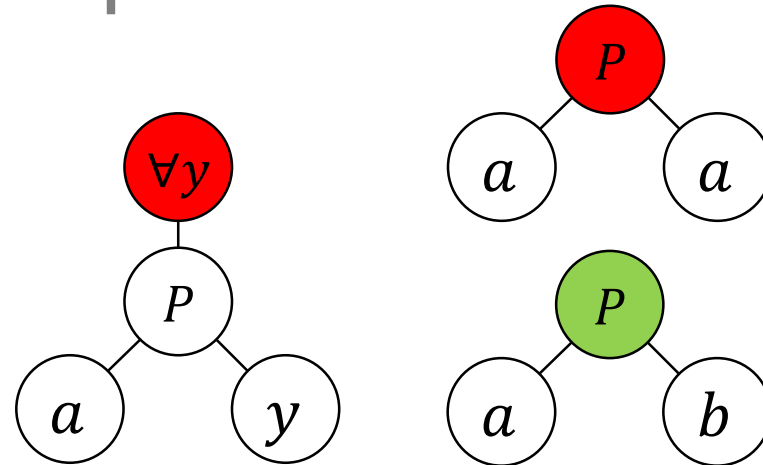
Example I

- $\phi = \forall x \exists y. P(x, y)$
- M :
 - $A = \{a, b\}$
 - $P^M = \{(a, b), (b, a)\}$



Example II

- $\psi = \exists x \forall y. P(x, y)$
- M :
 - $A = \{a, b\}$
 - $P^M = \{(a, b), (b, a)\}$



Example III

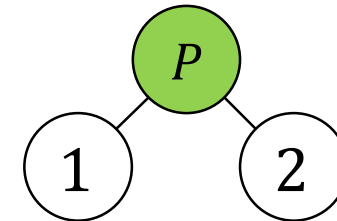
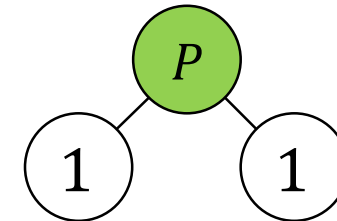
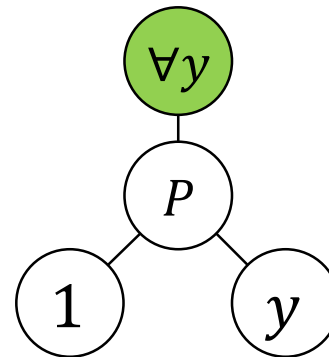
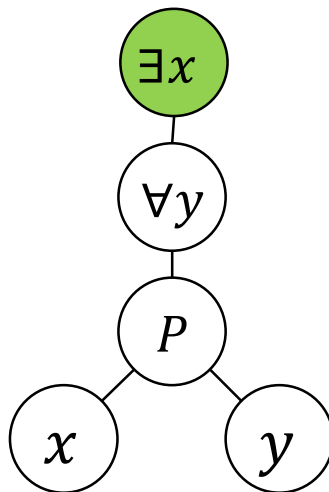
- $\psi = \exists x \forall y. P(x, y)$

- M :

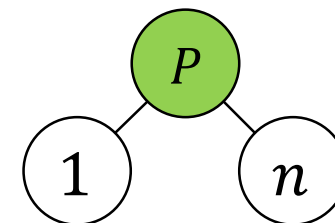
- $A = \mathbb{N}$

- $P^M = \leq =$

- $\{(1,1), (1,2), \dots, (2,2), \dots\}$



⋮



Semantic Entailment

- $\phi_1, \phi_2, \phi_3, \dots \models \psi$ iff:
 - For all models M
 - $M \models \phi$ for all $\phi_1, \phi_2, \phi_3, \dots$, then also $M \models \psi$

Validity

- ψ is **valid** iff:
 - For all models M
 - $M \models \psi$

Satisfiability

- ψ is **satisfiable** iff:
 - One model M
 - $M \models \psi$

Semantics of Equality

- Special case “=”
 - Denotes “real” equality
 - $=^M = \{(a_1, a_1), (a_2, a_2), (a_3, a_3), \dots\}$
 - In *all* models

Summary

- Syntax
 - Predicates
 - Functions
 - Quantifiers
- Semantics
 - Models
 - How to evaluate
- Validity, Satisfiability

