# Predicate Logic

Narges Khakpour

Department of Computer Science,
Linnaeus University

# Table of Content

# The limitations of propositional logic

- How should we express the followings in propositional logic?
  - Every student is younger than some instructor.
  - There is a student who is younger than an instructor.
  - Not all birds can fly.
  - Everybody has a father and a mother.
- Do they capture the true meaning of the sentences?
- E.g., in the first sentence, it's about being student, being younger, being an instructor, it is about all students etc
- Propositional logic dealt with sentence components like not, and, or and if ... then
- Cannot handle the logical aspects of natural and artificial languages.

# Predicate Logic

- Predicate logic assumes that the world consists of individual objects and the relations among them.
- Separate objects gives the possibility to quantify over them and make statements about all or some
- Adds functions, predicates and quantifiers to propositional logic

## Example

We define the following predicates:

- $Student(x)$ : $x$ is a student
- $Instructor(x)$ : $x$ is an instructor
- $Younger(x, y)$ : $x$ is younger than $y$.

# Table of Content

# Terms

- Constants: represent a specific object
- Variables: model objects of a specific domain (type), not always Boolean
- Functions: define properties of or relations among objects

## Definition

Let $\mathcal{F}$ be a set of functions. A term is defined as follows:

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

# Terms

## Example

- Let $\mathcal{F} = \{f_1, f_2, age\} \cup F_{arith}$ where $F_{arith}$ is the set of arithmetic functions.
- $f_1 = Alice,\ f_2 = Bob,\ age : \{Alice, Bob, John, Emma\} \implies \mathbb{N}$
- Alice is younger than Bob: $age(Alice) < age(Bob)$
- What are the (sub-)terms?

# Predicate

- To represent properties of object or their relations
- A function with a boolean co-domain, i.e. it returns true or false
- Consists of two parts
  - a name
  - list of arguments that are terms

## Example

Define a predicate to express "Alice is younger than Bob"!

$$Younger(Alice, Bob)$$

# Quantifiers

- Universal quantifier
  - the property is satisfied by ALL members of a group
  - Of the form $\forall a.\phi$ where $a$ is a variable and $\phi$ is a formula (discussed later)

Example

$\forall a.person(a) \implies has\_father(a)$

# Quantifiers

- Existential quantifier
  - AT LEAST ONE member of the group satisfies the property
  - Of the form $\exists a.\phi$ where $a$ is a variable and $\phi$ is a formula

**Example**

$person(Alice) \land \exists b.(person(b) \land is\_father\_of(Alice, b))$

BP How do we express "everybody has ONE father" using the predicate $is\_father\_of$.

# Logical Formulas

### Definition (Formula)

[1] A formula over $(\mathcal{F}, \mathcal{P})$ is defined as follows inductively

- If $P \in \mathcal{P}$ is a predicate symbol of arity $n \geq 1$, and if $t_1, t_2, ..., t_n$ are terms over $\mathcal{F}$, then $P(t_1, t_2, \ldots, tn)$ is a formula.
- If $\phi$ is a formula, then so is $(\neg\phi)$
- If $\phi$ and $\psi$ are two formulas, then so are $\phi \vee \psi$, $\phi \wedge \psi$ and $\phi \implies \psi$
- If $\phi$ is a formula and $x$ is a variable, then $(\forall x.\phi)$ and $\exists x.\phi$ are formulas.
- Nothing else is a formula.

# Logical Formulas Cont.

- If we assign each variable with a constant, the predicate becomes a proposition, $younger(Alice, Bob) \equiv younger_{AB}$.
- Operator precedence:
  Quantifiers $>$ Negation $>$ Conjunction $>$ Disjunction $>$ Implication $>$ Equivalence

## Example

- $\neg P(x) \wedge Q(x)$
- $\forall x.P(x) \wedge Q(x)$
- $\neg \forall x.P(x) \vee \forall x.Q(x)$

# Example

## Example

- Objects: students, teachers, courses
- Predicates
  - $Course(x)$: $x$ is a course
  - $teaches(t, c)$: $t$ teaches the course $c$
  - $teacher(t)$:$t$ is a teacher and $student(x)$ means $x$ is a student
  - $registerfor(x, y)$: $x$ has registered for the course $y$

# Example

## Example

- $1DV517$ is a course: $course(1DV517)$
- There should be at least one student registered to each course: $\forall c.(course(c) \implies (\exists s.student(s) \wedge registerfor(s, c)))$
- Why $\forall c.(course(c) \implies (\exists s.(student(s) \implies registerfor(s, c))))$ and $\forall c.(course(c) \wedge (\exists s.(student(s) \wedge registerfor(s, c))))$ don't specify the above?
- There are students who registered for $1DV517$: $\exists s.(student(s) \wedge registerfor(s, 1DV517))$

# Example

- The terms (in bold that are the predicates arguments)
  $\exists s.(student(\mathbf{s}) \wedge registerfor(\mathbf{s}, \mathbf{1DV517}))$
  $\forall c.(Course(\mathbf{c}) \implies (\exists s.(student(\mathbf{s}) \wedge registerfor(\mathbf{s}, \mathbf{c}))))$
- Which formula is a proposition?

# Quantifier Binding

- A **bound variable** either has a constant value or is quantified over.
- A variable that is not bound is **free**.
- An expression without free variables is called a **ground formula**, e.g.,$\forall x.((x > 1) \implies (x > 0))$
- Let $x$ be an individual variable. In expressions of the form $\exists x \phi$ or $\forall x.\phi$, the scope of quantifier is $\phi$
- In $\forall x.\phi$ or $\exists x \phi$, the quantifier binds every occurrence of $x$ in $\phi$, unless it is bounded by an *inner quantifier*.

# Quantifier Binding- Example

**Example**

Which variables are bound?

- $\forall x.student(x) \wedge happy(x)$
- $\forall x.(student(x) \wedge happy(x))$
- $\forall x.(student(x) \wedge \exists x.happy(x))$
- $\exists x.happy(y)$
- $happy(y)$

# Table of Content

# Tautology

- **A predicate formula is tautology if it is true for all possible interpretations.**
- Examples
  - $p(x) \vee \neg p(x)$
  - $\forall x.\forall y.(p(x,y) \vee \neg p(x,y))$
  - $p(x,y) \vee q(x) \equiv q(x) \vee p(x,y)$
  - $p(x) \vee \neg p(x) \equiv q(y) \vee \neg q(y)$
- We use the proof (inference) rules introduced for propositional logic
- Extra proof rules are required to reason about quantifiers

# Substitution

- Replacing a **free** variable by another valid term
- For example, let the formula $\phi$ be $P(x, y) \implies \neg Q(x, y)$
- $x$ and $y$ are free variables of type number
- Substituting $x$ with $2a * b$ leads to $P(2a * b, y) \implies \neg Q(2a * b, y)$
- This substitution is represented by $\phi[2a * b/x]$
- **We CANNOT substitute bound variables or constants, e.g. we cannot substitute $x$ in $\forall x.(P(x, y) \implies \neg Q(x, y))$.**

# Inference Rules for Equality

- The first rule:

$$\frac{}{t = t}$$

- Let $\phi$ be a tautology with a free variable $r$
- $\phi[t/r]$ is an expression obtained by substituting all occurences of $r$ with the term $t$

$$\frac{t_1 = t_2 \quad \phi[t_1/x]}{\phi[t_2/x]}$$

# Inference Rules for Universal Quantifiers

- Universal instantiation ($t$ is a term)

$$\frac{\forall x \; p(x)}{p(t/x)}$$

- Universal generalization

$$\frac{\boxed{\begin{array}{l} x_0 \\ \qquad \vdots \\ \qquad \phi[x_0/x] \end{array}}}{\forall x \, \phi}$$

where the box shows the scope of temporary variable $x_0$ (and not any assumption like what we had in the proof theory of propositional logic)

# Example

Prove $\forall x.(P(x) \wedge \neg Q(x)), \forall x.P(x) \vdash \forall x.Q(x)$

| | | | |
|---|---|---|---|
| 1 | $\forall x\,(P(x) \to Q(x))$ | premise | |
| 2 | $\forall x\,P(x)$ | premise | |
| 3 | $x_0$ | | |
| 4 | $P(x_0) \to Q(x_0)$ | $\forall x$ e | 1 |
| 5 | $P(x_0)$ | $\forall x$ e | 2 |
| 6 | $Q(x_0)$ | $\to$ e | 4, 5 |
| 7 | $\forall x\,Q(x)$ | $\forall x$ i | 3-6 |

# Inference Rules for Existential Quantifiers

- Existential generalization ($t$ is a term)

$$\frac{p(t/x)}{\exists x\ p(x)}$$

- Existential instantiation

$$\dfrac{\exists x\, \phi \qquad \boxed{\begin{array}{l} x_0\ \phi[x_0/x] \\ \qquad \vdots \\ \qquad \chi \end{array}}}{\chi}\ \exists\mathrm{e}.$$

$\chi$ is an $x_0$-free formula

# Example

Prove $\forall x.\ (P(x) \wedge \neg Q(x)), \exists x\ P(x) \vdash \exists x\ Q(x)$

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

## Quantifiers Equivalences

- $$\begin{aligned} \neg\forall x.P(x) &\not\equiv \neg p(x1) \wedge \neg p(x2) \wedge \ldots \wedge \neg p(xn) \\ &\equiv \neg(p(x1) \wedge p(x2) \wedge \ldots \wedge p(xn)) \\ &\equiv \neg p(x1) \vee \neg p(x2) \vee \ldots \vee \neg p(xn) \\ &\equiv \exists X.\neg p(X) \end{aligned}$$
- In general $\neg\forall x.\ \phi \equiv \exists x \neg\phi$:
  "$\phi$ is not true for all $x$ iff there is a $x$ that makes it false"
- Similarly $\neg\exists x.\ \phi \equiv \forall x(\neg\phi)$:
  "There is no $x$ that makes $\phi$ true iff it is false for all $x$"

# Quantifiers Equivalences

- If $x$ is not free in $\phi$:

$$\begin{aligned} \phi \wedge \forall x \psi &\equiv \forall x(\phi \wedge \psi) \\ \phi \vee \forall x \psi &\equiv \forall x(\phi \vee \psi) \\ \phi \wedge \exists x \psi &\equiv \exists x(\phi \wedge \psi) \\ \phi \vee \exists x \psi &\equiv \exists x(\phi \vee \psi) \end{aligned}$$

  why?

- Multiple quantifiers:

$$\forall x \forall y \phi \equiv \forall y \forall x \phi$$

$$\exists x \exists y \phi \equiv \exists y \exists x \phi$$

# Existential Proof

- How can we prove $\exists x \; p(x)$?
- Method 1: Constructive proof, i.e. find $c$ such that $p(c)$ holds
- Method 2: Proof by contradiction, i.e. prove that $\forall x \; \neg p(x) \implies \bot$

## Example- Constructive Proof

Prove that

$$[\forall X.(p(X) \implies q(X)), p(d), s(d)] \vdash \exists X.(q(X) \wedge s(X))$$

$d$ is a constant/instance

$$\begin{array}{lll}
1. & \forall X.(p(X) \implies q(X)) & \textit{Premise} \\
2. & p(d) & \textit{Premise} \\
3. & s(d) & \textit{Premise} \\
4. & p(d) \implies q(d) & \textit{Universal instantiation}: 1 \\
5. & q(d) & \textit{Modus ponens}: 2\&4 \\
6. & q(d) \wedge s(d) & 3\&5 \\
7. & \exists X.(q(X) \wedge s(X)) & \textit{Existential generalization}
\end{array}$$

# Example- Proof By Contradiction

Prove $\neg\forall x.\ \phi \vdash \exists x.\ \neg\phi$

| | | | |
|---|---|---|---|
| 1 | $\neg\forall x\, \phi$ | premise | |
| 2 | $\neg\exists x\, \neg\phi$ | assumption | |
| 3 | $x_0$ | | |
| 4 | $\neg\phi[x_0/x]$ | assumption | |
| 5 | $\exists x\, \neg\phi$ | $\exists x$ i | 4 |
| 6 | $\bot$ | $\neg$ e | 2, 5 |
| 7 | $\phi[x_0/x]$ | PBC | 4-6 |
| 8 | $\forall x\, \phi$ | $\forall x$ i | 3-7 |
| 9 | $\bot$ | $\neg$ e | 1, 8 |
| 10 | $\exists x\, \neg\phi$ | PBC | 2-9 |

# Proof by Induction

- To prove that a formula $P(n)$ is true for every natural number $n$
- **Base Case** Prove that it holds for $n = 0$
- **Inductive Hypothesis** Assume that it holds for any $n = k$, i.e., $P(k)$ is true.
- **Inductive Step** Prove that it holds for $n = k + 1$, i.e., $P(k + 1)$ is true

# Example

- Prove that $2k \leq 2^k$, $1 \leq k$
- **Base Case** $2 \leq 2$: it holds.
- **Inductive Hypothesis** Assume that $2n \leq 2^n$ holds.
- **Inductive Step** Prove $2(n+1) \leq 2^{n+1}$

$$
\begin{array}{ll}
2n \times 2 \leq 2^n \times 2 & \text{From Inductive Step Hypothes} \\
\implies 4n \leq 2^{(n+1)} & (1) \\
\text{And} & \\
1 \leq n \implies 2 \leq 2n \implies & \\
0 \leq 2n - 2 \implies 0 \leq 4n - 2n - 2 & \\
\implies 2n + 2 \leq 4n \implies 2(n+1) \leq 4n & (2) \\
\stackrel{(1),(2)}{\implies} 2(n+1) \leq 4n \leq 2^{(n+1)} &
\end{array}
$$

# Limitations of predicate logic

- Can we express everything using predicate logic?
  - “Most students will pass the exam”
  - “The car is very fast”
  - ”He studied the book before attending the lecture.”
- Other types of logic
  - Second order predicate logic (predicates as arguments to predicates)
  - Fuzzy/probabilistic logic (approximate statements)
  - Modal logics (Epistemic logic, Temporal logic, deontic logic etc)

# Table of Content

# Summary

- Predicate logic is a more expressive logic to express knowledge
- Syntax of predicate logic was introduced
- Proof in predicate logic was discussed