Introduction to Predicate Logic

Part 1

Professor Anita Wasilewska

Lecture Notes (1)
Introduction

• Lecture Notes (1) and (2) provide an OVERVIEW of a standard intuitive formalization and introduction to Predicate Logic.

• I wrote them for you to review it and use it to solve some of the Take–home Practice Final problems.

• It covers material that I assume is KNOWN to all students;

• The formal definitions are in Chapter 13, 14 and we will cover it later.
Predicate Logic Language

Symbols:
1. P, Q, R... predicates symbols, denote relations in “real life”, must be a non empty set. It can be finite or countably infinite.
2. x, y, z.... variables, a countably infinite set
3. c_1, c_2, ... constants, a finite or countably infinite set
4. f, g, h ... functional symbols, denote functions in “real life”, may be empty, finite, or countably infinite set

5. Propositional connectives:
   \[ \lor, \land, \Rightarrow, \neg, \iff \]

6. Symbols for quantifiers
   \[ \forall x \] – universal quantifier reads: For all x...
   \[ \exists x \] – existential quantifier reads: There is x...
Formulas of Predicate Logic

We use symbols 1 - 6 to build **formulas** of predicate logic as follows

1. P(x), Q(x,y), R(x)... R(c1), Q(x, c3),  P(c), ...
   are called **atomic formulas** for any variables x, y,... and constants c, c1, c2, ...

2. All atomic formulas are formulas ;

3. If A,B are formulas then (like in propositional logic):
   (A ∨ B) , (A ∧ B), (A ⇒ B), (A ⇔ B), ¬A
   are **formulas**

4. ∀x A,  ∃ y A  are **formulas**, for any variables x, y

5. The set \( \mathcal{F} \) of **all formulas** is the **smallest** set that fulfills the conditions 1 -4.
Examples

For example: let
P(y), Q(x,c), R(z), P_1(x, y, z) be \textbf{atomic} formulas, i.e.

\[ P(x), Q(x,c), R(z), P_1(x, y, z) \in \mathcal{F} \]

Then we form some other formulas out of them as follows:

\[ (P(y) \lor \neg Q( x, c)) \in \mathcal{F} \]

It is a \textbf{formula} with two \textbf{free variables} x, y.

\[ \exists x (P(y) \lor \neg Q( x, c)) \in \mathcal{F} \]

\[ \forall y (P(y) \lor \neg Q( x, c)) \in \mathcal{F} \]

\[ \forall y \exists x (P(y) \lor \neg Q( x, c)) \in \mathcal{F} \]

etc
Free and Bound Variables

Quantifiers **bound** variables within formulas

For example: A is a formula:

$$\exists x \ (P(x) \Rightarrow \neg Q(x, y))$$

all x’s in A are bounded by $$\exists x$$

y is a free variable in A that **can be bounded** by a quantifier, for example

$$\forall y \ \exists x \ (P(x) \Rightarrow \neg Q(x, y))$$

y got bounded and there are no free variables in A now.

A formula without free variables is called a sentence.
Logic and Mathematical Formulas

We often use logic symbols while writing mathematical statements in a more symbolic way.

**Example of a Mathematical Statement:**

\[ \forall x \in \mathbb{N} \ (x > 0 \land \exists y \in \mathbb{N} \ (y = 1)) \]

1. **Quantifier** \( \forall x \in \mathbb{N} \) is a quantifier with restricted domain.
2. **Logic uses only** \( \forall x, \exists y \)
3. **\( x > 0 \) and \( y = 1 \)** are mathematical statement about a real relation “>”
4. **Logic uses symbols** \( P, Q, R \ldots \) for example \( R(y, c_1) \) for \( y = 1 \) and \( P(x, c_2) \) for \( x > 0 \) where \( c_1 \) and \( c_2 \) are constants representing numbers 1 and 0, respectively.
Translation of Mathematical Statements to Logic

Consider a Mathematical Statement:

\[ \forall x \in \mathbb{N} \ (x > 0 \land \exists y \in \mathbb{N} \ (y = 1)) \]

\( x \in \mathbb{N} \) – we translate it as one argument predicate \( Q(x) \)
\( x > 0 \) – as two argument predicate \( P(x, c_1) \), \( y = 1 \) – as another two argument predicate \( R(y, c_2) \) and get

\[ \forall Q(x) \ (P(x, c_1) \land \exists Q(y) \ R(y, c_2)) \]

↑ Logic formula with restricted domain quantifiers

But this is not yet a proper logic formula since we cannot have quantifiers \( \forall Q(x) \), \( \exists Q(y) \) but only \( \forall x, \exists x \).

\( \forall Q(x), \exists Q(y) \) are called quantifiers with restricted domain
Logic Formula Corresponding to our Mathematical Statement

We need to “get rid” of quantifiers with restricted domain i.e. to translate them into logic quantifiers: \( \forall x, \exists x \)

\( \exists x \in \mathbb{N}, \exists y \in \mathbb{N} \)

\( \uparrow \) certain predicate \( P(x) \)

General: restricted domain quantifiers are:

\( \forall P(x), \exists Q(x) \)
Restricted Domain Existential Quantifier

\[ \exists_{P(x)} Q(x) \equiv \exists x (P(x) \land Q(x)) \]

↑ restricted    ↑ logic, not restricted

Example (mathematical formulas):
\[ \exists x \neq 1 \ (x > 0 \Rightarrow x + y > 5) \text{ - restricted} \]
\[ \exists x ((x \neq 1) \land (x > 0 \Rightarrow x + y > 5)) \text{ - not restricted} \]

↑ P(x, y, c)

English statement:
Some students are good.

Logic Translation (restricted domain):
\[ \exists_{S(x)} G(x) \]

Predicates are:
S(x) – x is a student
G(x) – x is good

TRANSLATION:
\[ \exists x (S(x) \land G(x)) \]
Restricted Quantifiers and Quantifiers

Translation for universal quantifier

Restricted Logic (Non-restricted)

\[ \forall_{P(x)} Q(x) \equiv \forall x (P(x) \Rightarrow Q(x)) \]

Example (mathematical)

\[ \forall x \in \mathbb{N} (x = 1 \lor x < 0) \quad \text{restricted domain} \]

\[ \equiv \forall x (x \in \mathbb{N} \Rightarrow (x = 1 \lor x < 0)) \quad \text{– non-restricted} \]
Translation of Mathematic statements to Logic formulas

Mathematical statement:
\[ \forall x \left( \forall x \in N \Rightarrow (x = 1 \vee x < 0) \right) \]
x \in N – translates to a predicate  \( N(x) \)
x > 0 – translates to a predicate  \( P(x, c_1) \)
x=1 – translates to a predicate  \( Q(x, c_2) \)

Corresponding logic formula:
\[ \forall x \left( N(x) \Rightarrow (Q(x, c_2) \vee P(x, c_1)) \right) \]
Remark

Mathematical statement: \( x + y = 5 \)

To follow LOGIC formalization we re-write it as \( = ( + (x, y), 5) \)

and “translate it” to Predicate Language as \( P(f(x,y), c) \)

Given \( x = 2, x = 1 \), we get \( +(2,1) = 3 \) and the statement:
\( = (3,5) \) is FALSE \( (F) \)

**Predicates always returns logical value F or T**

We really need also **function** symbols (like +, etc..) to translate mathematical statements to logic, even if we could use only relations as functions are special relations

This is why in **formal** definition of the predicate language we often we have 2 sets of symbols

1. **Predicate** symbols which can be true or false in proper domains

2. **Functions** symbols (formally called **terms**)
Translations to Logic

Rules:
1. **Identify** the domain: always as et X ≠ φ
2. **Identify** predicates (simple: atomic)
3. **Identify** functions (if needed)

4. **Identify** the connectives V, ∧, ⇒, ¬, ⇔
5. **Identify** the quantifiers ∀x, ∃x

Write a formula using only symbols for 2, 3, 4

6. **Use restricted domain quantifier translation rules**, where needed
Translations Examples

Translate:
For every bird there are some birds that are white

Predicates:
- B(x) – x is a bird
- W(x) – x is white

Restricted:
\[ \forall_{B(x)} \exists_{B(x)} W(x) \]

Logic:
\[ \forall x (B(x) \Rightarrow \exists x (B(x) \land W(x))) \]

OR (re-name variables)
\[ \forall x (B(x) \Rightarrow \exists y (B(y) \land W(y))) \]
(by laws of Quantifiers, don’t do this step yet!! We will study the laws later)

This is logically equivalent to
\[ \forall x \exists y (B(x) \Rightarrow (B(y) \land W(y))) \]
Example

For every student there is a student that is an elephant

B(x) - x is a student

W(x) – x is an elephant

\( \forall_{B(x)} \exists_{B(x)} W(x) \)

\( \forall_{B(x)} \exists x (B(x) \land W(x)) \)

\( \forall x (B(x) \Rightarrow \exists x (B(x) \land W(x))) \)  \hspace{1em} (logic formula)
Translate: *Some patients like all doctors*

**Predicates:**
- $P(x) – x$ is a patient
- $D(x) – x$ is a doctor
- $L(x, y) – x$ likes $y$

$\exists P(x) \ \forall D(y) \ L(x, y)$

There is a **patient**($x$), such that for all **doctors**($y$), $x$ likes $y$

$\exists x(P(x) \land \forall y(D(y) \Rightarrow L(x, y)))$

(by law of quantifiers to be studied later we can “pull out $\forall y$”)

$\exists x\forall y(P(x) \land (D(y) \Rightarrow L(x, y)))$