Predicate Logic
(part 1)
(Chapter 2)

CSE 352 Artificial Intelligence

Professor Anita Wasilewska

Lecture Notes (3)

Predicate Logic Language

Symbols:

1. P, Q, R… predicates symbols, denote relations in "real life", countably infinite set
2. x,y,z…. variables, countably infinite set
3. c1, c2, … constants, countably infinite set
4. f, g, h … functional symbols, may be empty, denote functions in "real life"
5. **Propositional connectives:**

   ∨, ∧ , ⇒, ¬, ⇔

6. **Symbols for quantifiers**

   ∀x – universal quantifier reads: For all x…

   ∃ 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):

   $(A \vee B)$, $(A \wedge B)$, $(A \Rightarrow B)$, $(A \Leftrightarrow B)$, $\neg A$

   are **formulas**

4. **$\forall x$ A, $\exists$ y A** are **formulas,** for any variables x, y
5. The set $\mathscr{F}$ 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 **atomic** formulas, i.e.

$P(x), Q(x,c), R(z), P_1(x, y, z) \in \mathscr{F}$

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

$(P(y) \vee \neg Q(x, c)) \in \mathscr{F}$

It is a **formula** with two **free variables** x, y.

$\exists x\ (P(y) \vee \neg Q(x, c)) \in \mathscr{F}$

$\forall y\ (P(y) \vee \neg Q(x, c)) \in \mathscr{F}$

$\forall y\ \exists x(P(y) \vee \neg Q(x, c)) \in \mathscr{F}$

etc

Free and Bound Variables

Quantifiers **bound** variables within formulas

For example: A is a formula:

∃**x** (P(**x**) ⇒ ¬ Q(**x**, y))

all the **x**'s in A are **bounded** by **∃x**

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

∀**y** ∃x (P(x) ⇒ ¬ 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:

∀x ∈ N (x > 0 ∧ ∃y ∈ N (y = 1))

1. Quantifier **∀x ∈ N** is a quantifier with **restricted domain**
2. **Logic uses only ∀x , ∃y**
3. **x > 0 and y =1** are mathematical statement about "real relation" **>**
4. Logic **uses symbols** P, Q, R… 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:

$\forall x \in N\ (x > 0 \land \exists y \in N\ (y = 1))$

$x \in N$ – we translate it as one argument predicate **$Q(x)$**

$x > 0$ – as **$P(x, c_1)$** , **$y = 1$** – as **$R(y, c_2)$** and get

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

↑ Logic formula

But this is not yet a proper 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: $\forall x$, $\exists x$**

$\exists x \in N$, $\exists y \in N$ are restricted quantifiers

↑ certain **predicate** P(x)

General: restricted domain quantifiers are :

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

Restricted Domain Existential Quantifier

$$\exists_{P(x)} Q(x) \equiv \exists x(P(x) \wedge Q(x))$$

↑ **restricted** ↑**logic, not restricted**

Example (mathematical formulas):

$\exists\ x\neq1\ (x>0 \Rightarrow x + y > 5)$ - restricted

$\exists x\ ((x\neq1\ ) \wedge (x > 0 \Rightarrow x + y > 5))$ - 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:

$$\exists x(S(X) \wedge G(x))$$

Restricted Quantifiers and Logic Quantifiers

Translation for universal quantifier

Restricted Logic (Non-restricted)

$\forall_{P(x)} Q(x) \quad \Xi \quad \forall x\, (P(x) \Rightarrow Q(x))$

Example (mathematical)

$\forall x \in N\, (x = 1 \lor x < 0)$ restricted domain

$\Xi\ \forall x\, (x \in N \Rightarrow (x = 1 \lor x < 0))$ – non-restricted

Translation of Mathematic statements to Logic formulas

Mathematical statement:

$\forall x\ (x \in N \Rightarrow (\ x = 1 \vee x < 0)\ )$

$x \in N$ – translates to $N(x)$

$x < 0$ – translates to $P(x, c_1)$

$x < y$ – $<$ is a 2 argument relation ≡ two argument predicate $P(x, y)$, x, y are variables

0 – constant – denote by $\mathbf{c_1}$

$x=1$ - $=$ is a two argument predicate $Q(x,y)$,

$x = 1$, - 1 is constant denoted by c_2,

$x=1$ translates to $Q(x, c_2)$

Corresponding logic formula:

$\forall x\ (N(x) \Rightarrow (Q(x, c_2) \vee P(x, c_1))\ )$

Remark

Mathematical statement: **x + y = 5**

We re-write it as

= (+ (x, y), 5)

Given x = 2, x = 1, we get **+(2,1) = 3** and the statement:

= (3,5) is FALSE (F)

Predicates always returns 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 **∨, ∧ , ⇒, ¬, ⇔**
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 we can re-write it as

$\forall x \exists y\,(B(x) \Rightarrow (B(y) \land W(y)))$

Example

For every student there is a student that is an elephant

B(x)- x is a student

W(x) – x is an elephant

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

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

$\forall x(B(x) \Rightarrow \exists x(B(x) \land W(x)))$ (logic formula)

Translations Example

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

$\boldsymbol{\exists_{P(x)} \forall_{D(y)} L(x,y)}$

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

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

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

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