Discrete Mathematics and Statistics - CPT107

Xi'an Jiaotong-Liverpool University
西交利物浦大学

Predicate Logic

The Need for a Richer Logic

Predicate logic is an extension of Propositional logic.

Contents

- Predicates and Quantifiers
- Translating English into predicate logic

Limitations of Propositional Logic

- Propositional logic is good for dealing with statements built from basic propositions using *and*, *or*, *not*, *if … then*

> Example:
>
> *“**If** a student hands in his or her assignment late*
> ***and** there is a good reason for the late submission,*
> ***then** there will be no penalty.”*

- More complex statements cannot be expressed naturally in propositional logic

> Example:
>
> *“Every student is younger than some instructor.”*

Another Example: Cash Machines

- Propositional logic can express the following:

> *“**If** the debit card is inserted*
> ***and** the PIN is locked,*
> ***then** access to the account is denied.”*

- But it does not help us to express:

> *“Each cash withdrawal is limited*
> *by the acount’s balance.”*

The Need for a Richer Logic

- We need to extend propositional logic.
- Numerous extensions have been developed, among them:
  - Modal Logic
    Adds elements to reason about necessity.
  - Temporal Logic
    Adds elements to reason about time.
  - Predicate Logic (a.k.a. First-order Logic):
    Adds elements to reason about properties of objects.

Predicate logic

- is to express any common or mathematical statements and therefore provides an adequate framework for logical reasoning.

Applications of Predicate Logic

- It is *the* formal notation for writing perfectly clear, concise, and unambiguous mathematical *definitions*, *axioms*, and *theorems* for *any* branch of mathematics.
- Supported by some of the more sophisticated *database query engines.*
- Basis for *automatic theorem provers* and many other *Artificial Intelligence* systems.

Predicate logic

Predicate logic has two major aspects:

- precise *syntax* involving a formal language, called a *first-order language*, that enables us to express statements in a uniform way, by means of formulae;

- formal *semantics*, specifying the meaning of all components of the language by means of their interpretation in a mathematical structure, and formal truth definitions, extending the truth tables for the propositional connectives.

Predicates

- Recall the statement:

> “Every student is younger than some instructor.”

- What is it about?
  - Being a student
  - Being younger than somebody else
  - Being an instructor

these are properties

- Idea: use predicates to express such properties

Example

Predicates:

- Student: is a student
- Younger: is younger than
- Instructor: is an instructor

statement	**expressed using predicates**
Alice is a student	*Student*(*alice*)
Alice is younger than Bob	*Younger*(*alice*, *bob*)
Bob is an instructor	*Instructor*(*bob*)

Predicate

Predicate

- Object x has property P : P (x)
- Relation R holds between object x and object y: R(x, y)

Q(x_1, x_2, . . . , x_n)

- Q holds for objects x_1, x_2, . . . , x_n
- Q is a predicate with *n* variables.

Student(x) = “*x* is a student“

- *Student* (alice) = “Alice is a student"

Instructor(y) = “y is an instructor”

- *Instructor* (bob) = “Bob is an instructor"

Younger(x, y) = “*x* is younger than *y*"

- *Younger*(alice, bob) = “Alice is younger than Bob "

Predicates are Not Enough

- Predicates are not yet enough to express our statement:

> “Every student is younger than some instructor.”

- We need a mechanism to:
  - Express “***every …***” and “***some …***”.
  - Talk about students and instructors without mentioning each one by name.

    (The above statement doesn’t really care about the names of students and instructors, just the connection between students and instructors.)

Variables to the Rescue

Idea: use variables

- as place holders for concrete values (like students, instructors, account numbers, etc.)
- Notation: x, y, z, … (possibly with decorations like subscripts)

Examples:

$\mathit{Student}(x)$	… “x is a student”
$\mathit{Younger}(\mathit{alice}, y)$	… “Alice is younger than y”
$\mathit{Younger}(z_1, x)$	… “z_1 is younger than x”
$\mathit{Instructor}(z)$	… “z is an instructor”

Quantifiers

Our original sentence:

> "Every student is younger than some instructor."

can be rephrased using predicates and variables as:

> "**For every** individual x with *Student*(x), **there is** an individual y such that *Younger*(x, y) and *Instructor*(y)."

To express "*for all*" and "*exists*", we use quantifiers $\forall$ and $\exists$:

- $\forall$ … for "for all" Universal quantification
- $\exists$ … for "there exists an" Existential quantification

The Statement in Predicate Logic

The original statement can now be written symbolically using predicates, variables, and quantifiers:

$$\forall x \Big(Student(x) \rightarrow \exists y \left(Younger(x, y) \land Instructor(y) \right) \Big)$$

- This is a typical formula in predicate logic!
- Translating the formula back yields:

> *"For every individual x, if x is a student, then there is an individual y such that x is younger than y and y is an instructor."*

Translation into Predicate Logic II

"Not all birds can fly."

1. Pick suitable predicates: *Bird* (being a bird), *CanFly* (can fly)
2. Encode the sentence in predicate logic:

$$\neg\forall x\big(\mathit{Bird}(x) \rightarrow \mathit{CanFly}(x)\big)$$

"It is not the case that for all individuals *x*, if *x* is a bird, then *x* can fly."

Equivalently:

$$\exists x\big(\mathit{Bird}(x) \land \neg\mathit{CanFly}(x)\big)$$

"There is an individual that is a bird and that cannot fly."

This equivalence can be made precise (later lecture).

Equality and Function Symbols

Predicate logic has two additional features:

1. Equality: allows to express that individuals are equal or not

Example: *"At least two students are registered for COMP118."*

$$\exists x\, \exists y\, \big(\mathit{Registered}(x, \mathit{comp118}) \wedge \mathit{Registered}(y, \mathit{comp118}) \wedge \neg x = y \big)$$

2. Function symbols: allows to express functional dependencies between individuals

Example: *"Alice and Bob have the same mother."*

$$\mathit{Mother}(\mathit{alice}) = \mathit{Mother}(\mathit{bob})$$

Simulating Function Symbols by Predicates

- Function symbols can often be "simulated" by predicates, so they are not really necessary

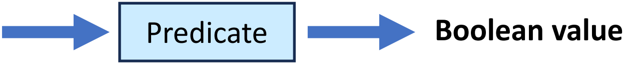
Example: $\mathit{Mother}(\mathit{alice}) = \mathit{Mother}(\mathit{bob})$ can be expressed as

$$\forall x \, \forall y \, \big(\mathit{IsMotherOf}(x, \mathit{alice}) \land \mathit{IsMotherOf}(y, \mathit{bob}) \to x = y \big)$$

- But function symbols lead to more natural descriptions

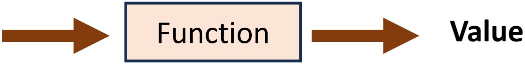
Example: compare the two formulae above

Predicate logic vs Function



Predicate is to show a comparison or a relation between two objects, such as:

Ex: IsMotherOf(x, alice) ; President(Obama, America)



Functions are to specify what a particular object is, such as:

Ex: Mother(alice) ; Human(Einstein)

A predicate is a function that returns true or false (Boolean function).

A function associates individuals and predicates their meanings.

The Language of Predicate Logic

Domains (Domains of discourse): a *non-empty set of objects* (humans, animals, account numbers, …).

- In *mathematics*, the domain usually consists of numbers (*integers, rationals, reals, etc.*), vectors, geometric figures (*points, lines, triangles, spheres, etc.*), sets, graphs, etc.
- A domain in a *nonmathematical* discourse may consist of material objects, human beings, ideas or anything else.

Constants: *concrete objects in the domain*. Some objects in the domain can be distinguished in a way that would allow us to make direct references to them, by giving them *names*.

e.g. in the domain of real numbers are 0, 1, π,e, etc.

The Language of Predicate Logic (II)

Variables: when we deal with *unknown or unspecified objects* from the domain of discourse.

- in natural language we usually use pronouns or other syntactic constructions

  Example "*If a man owes money to another man then that man hates the other man*"

- we use variables as *placeholders for the arguments of the various predicates and functions* we deal with.
- for variables we will be using *letters u, v, w, x, y, z,* possibly indexed.

  Example

  ○ the (unary) functions: f (x) = x + 1 and m(z) = "*the mother of z*",

  ○ the (binary) predicates x < y and "x loves y", etc.

The Language of Predicate Logic (III)

Predicates. A predicate takes object(s) in the domain as argument(s) and returns *true* or *false*.

They describe *properties* of objects or *relationships* between objects.

> ***Example*** talking about integers - properties like being "*positive*", "*divisible by 3*", "*not greater than 2000*", etc.

- ***unary*** predicates
- ***binary*** predicates - relating two objects: *"_is less than _"*, *"_ is a son of _"*, etc.
- ***n-ary*** predicates, relating *n* objects.

As long as we specify the meaning (*semantics*) of a predicate and the *objects* that it relates, that predicate becomes true or false.

We can connect predicates, by using propositional connectives

The Language of Predicate Logic (IV)

Function symbols. We use functions to represent *operations* that, applied to one or several objects, *determine an object*.

Depending on the number of arguments, we talk *about unary, binary, etc., functions.*

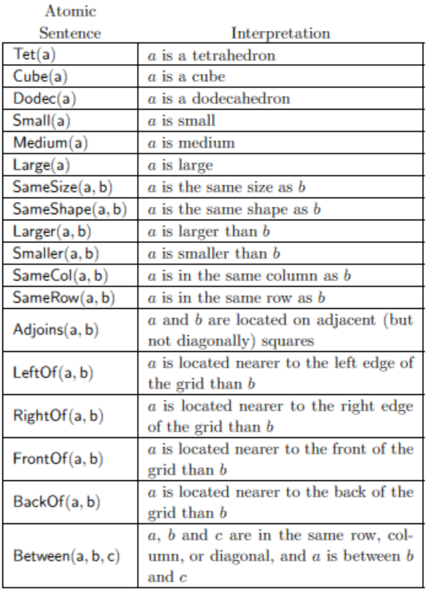
Example
“the mother of_”, “the older grandfather of_”, etc., in the domain of humans.

Logical symbols, including:

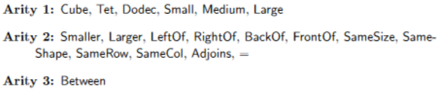
(a) **propositional connectives**,
(b) **quantifiers** - to quantify over objects of our discourse,

Auxiliary symbols, such as (,)

Example List of predicates



Atomic Sentence	Interpretation
Tet(a)	a is a tetrahedron
Cube(a)	a is a cube
Dodec(a)	a is a dodecahedron
Small(a)	a is small
Medium(a)	a is medium
Large(a)	a is large
SameSize(a, b)	a is the same size as b
SameShape(a, b)	a is the same shape as b
Larger(a, b)	a is larger than b
Smaller(a, b)	a is smaller than b
SameCol(a, b)	a is in the same column as b
SameRow(a, b)	a is in the same row as b
Adjoins(a, b)	a and b are located on adjacent (but not diagonally) squares
LeftOf(a, b)	a is located nearer to the left edge of the grid than b
RightOf(a, b)	a is located nearer to the right edge of the grid than b
FrontOf(a, b)	a is located nearer to the front of the grid than b
BackOf(a, b)	a is located nearer to the back of the grid than b
Between(a, b, c)	a, b and c are in the same row, column, or diagonal, and a is between b and c



Arity 1: Cube, Tet, Dodec, Small, Medium, Large

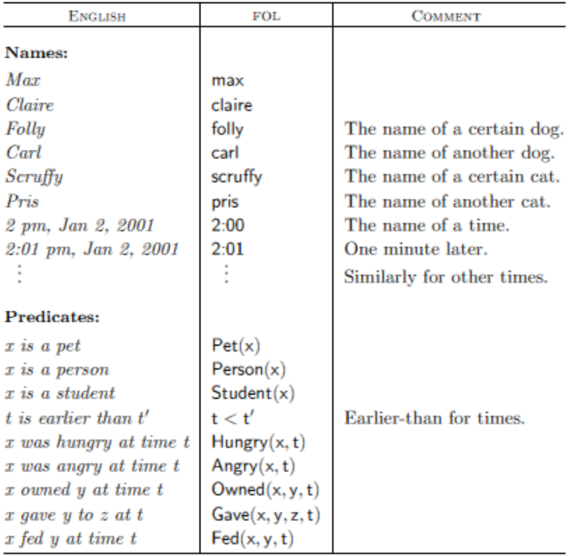
Arity 2: Smaller, Larger, LeftOf, RightOf, BackOf, FrontOf, SameSize, SameShape, SameRow, SameCol, Adjoins, =

Arity 3: Between

Example Names and predicates for a language

Concrete objects in the domain

To express some property of objects or some relation between objects.



English	FOL	Comment
Names:		
Max	max	
Claire	claire	
Folly	folly	The name of a certain dog.
Carl	carl	The name of another dog.
Scruffy	scruffy	The name of a certain cat.
Pris	pris	The name of another cat.
2 pm, Jan 2, 2001	2:00	The name of a time.
2:01 pm, Jan 2, 2001	2:01	One minute later.
⋮	⋮	Similarly for other times.
Predicates:		
x is a pet	Pet(x)	
x is a person	Person(x)	
x is a student	Student(x)	
t is earlier than t′	$t < t'$	Earlier-than for times.
x was hungry at time t	Hungry(x, t)	
x was angry at time t	Angry(x, t)	
x owned y at time t	Owned(x, y, t)	
x gave y to z at t	Gave(x, y, z, t)	
x fed y at time t	Fed(x, y, t)	

Syntax of predicate logic

Signatures

- A signature defines the “vocabulary” relative to which formulae can be expressed.
- It is the set of all the symbols that can be used in a formula:
  - Predicate symbols: *Student*, *Younger*, *Bird*, …
  - Constant symbols: *alice*, *comp118*, *42*, …
  - Function symbols: *Mother*, *Plus*, *Minus*, …
- It also fixes an arity (number of parameters) for each predicate and function symbol in it.

  Example: In the formulae of the last lecture, we assumed that *Student* and *Younger* have arity 1 and 2, respectively.

Signatures

Definition

A signature S is a set consisting of:

- predicate symbols, each with an associated arity
- function symbols, each with an associated arity
- constant symbols

Notation:

- $P, Q, R, \dots$, possibly with subscripts/superscripts, denote predicate symbols (unless otherwise stated)
- $F, G, \dots$ denote function symbols
- $a, b, c, \dots$ denote constant symbols

Example: Kinship Relations

A signature S_K for expressing kinship relations:

$$S_K = \{\mathit{Male}, \mathit{Female}, \mathit{Parent}, \mathit{Sibling}, \mathit{alice}, \mathit{Mother}, \mathit{Father}\},$$

where

- *Male* and *Female* are predicate symbols of arity 1
- *Parent* and *Sibling* are predicate symbols of arity 2
- *alice* is a constant symbol,
- *Mother* and *Father* are function symbols of arity 1

Note: We'll see later how to assign meaning to the symbols. For the moment, keep in mind that symbols do not have any meaning per se. The descriptive names above merely *hint at the intended meaning*.

Notation

We call a predicate or function symbol

- unary if its arity is 1
- binary if its arity is 2
- *k*-ary if its arity is *k*

Example: Arithmetic

A signature S_A for arithmetic:

$$S_A = \{\mathit{Smaller}, \mathit{0}, \mathit{1}, \mathit{Plus}, \mathit{Times}\},$$

where

- *Smaller* is a binary predicate symbol
- *0* and *1* are constant symbols,
- *Plus* and *Times* are binary function symbols

Example: Kinship Relations (

Signature $S = \{\mathit{alice}, \mathit{Mother}, \mathit{Father}\}$, where *alice* is a constant symbol, and $\mathit{Mother}, \mathit{Father}$ are unary function symbols

- *alice* can be used to refer to Alice.
- $\mathit{Mother}(\mathit{alice})$ can be used to refer to Alice’s mother.
- $\mathit{Father}(\mathit{Mother}(x))$ can be used to refer to *x*’s maternal grandfather.
- etc.

Example: Arithmetic

$S_A = \{\mathit{Smaller}, 0, 1, \mathit{Plus}, \mathit{Times}\}$

- x is an even number:

$$\exists y\, x = \mathit{Plus}(y, y)$$

- x is prime:

$$\mathit{Smaller}(1, x) \wedge \forall y \forall z \left(\mathit{Times}(y, z) = x \rightarrow (y = 1 \vee y = x) \right)$$

Important: This assumes that the universe of discourse is the set of natural numbers.

Semantics of Formulae

What does this formula mean?

$$\forall x\Big(\mathit{Student}(x) \to \exists y\,(\mathit{Younger}(x,y) \land \mathit{Instructor}(y))\Big)$$

- Every UoL student is younger than some instructor at UoL?
- Every student on earth is younger than some instructor at UoL's CS Department?
- Something else?

It has no meaning per se!

Semantics of Formulae

How do we assign meaning to formulae like:

$$\forall x \Big(\mathit{Student}(x) \rightarrow \exists y \left(\mathit{Younger}(x,y) \wedge \mathit{Instructor}(y) \right) \Big)$$

We have to specify:

1. The domain (of discourse).

   This is the set of all objects we are talking about (people, birds, integers, reals, etc.)

2. The meaning of all the symbols mentioned in the formula (predicate/function/constant symbols).

Example: Students and Instructors

$$\forall x \Big(Student(x) \rightarrow \exists y \left(Younger(x, y) \wedge Instructor(y) \right) \Big)$$

- A possible domain: $D = \{\text{Alice}, \text{Bob}, \text{Carol}\}$
- To say that Alice and Bob are students, we could interpret the predicate symbol *Student* with the set:

$$\{\text{Alice}, \text{Bob}\}$$

- To say that Alice is younger than Bob, and Bob is younger than Carol, we could interpret *Younger* with the set:

$$\{(\text{Alice}, \text{Bob}),\ (\text{Bob}, \text{Carol}),\ (\text{Alice}, \text{Carol})\}$$

Proposition logic vs Predicate logic

- in **Proposition logic**, we can express statements as a whole, and combinations of them. A proposition is a statement that is having a *truth value* (either true or false) associated with it.

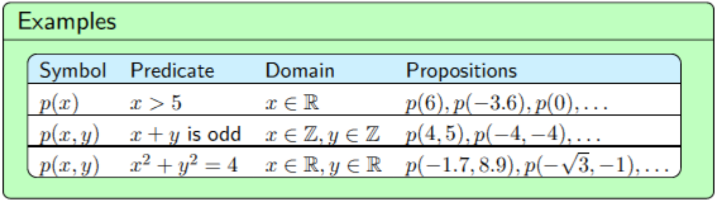
- in **Predicate Logic** (FOL) a statement has a specific inner structure, consisting of terms and predicates.

Terms (variable and functions) denote objects in some reality, and *predicates* express properties of, or relations between those objects.

The truth value is dependent upon the variables.

Proposition logic ***vs*** *Predicate logic*

- A predicate is a sentence that contains one or more variables
- A predicate is neither true nor false
- A predicate becomes a proposition when the variables are substituted with specific values
- The domain of a predicate variable is the **set of all values that may be substituted for the variable**



Examples

Symbol	Predicate	Domain	Propositions
$p(x)$	$x > 5$	$x \in \mathbb{R}$	$p(6), p(-3.6), p(0), \ldots$
$p(x, y)$	$x + y$ is odd	$x \in \mathbb{Z}, y \in \mathbb{Z}$	$p(4, 5), p(-4, -4), \ldots$
$p(x, y)$	$x^2 + y^2 = 4$	$x \in \mathbb{R}, y \in \mathbb{R}$	$p(-1.7, 8.9), p(-\sqrt{3}, -1), \ldots$

Xi'an Jiaotong-Liverpool University
西交利物浦大学

End of lecture

- **Summary**
  - Predicates and Quantifiers
  - Translating English into Predicate logic (FOL)

- **Reading**
  - Discrete Mathematics for Computer Scientists, J.K.Truss
  - Discrete Mathematics for Computing, R. Haggarty,