First Order Logic

Next Chapter:

Domain, Predicate

- By Deepak Poonia (IISc Bangalore)

Instructor:

Deepak Poonia
IISc Bangalore
GATE CSE AIR 53; AIR 67;
AIR 107; AIR 206; AIR 256

Discrete Mathematics Complete Course:

https://www.goclasses.in/courses/Discrete-Mathematics-Course

Join GO Classes **GATE CSE Complete Course** now:

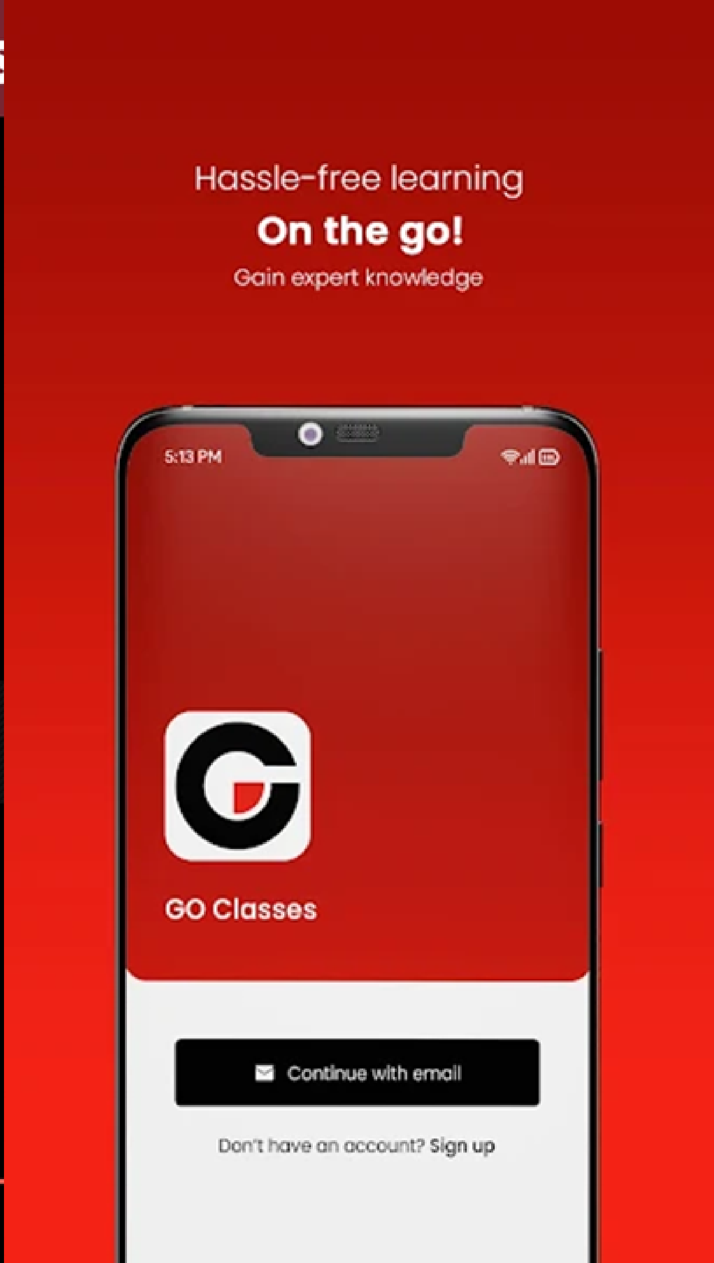
https://www.goclasses.in/s/pages/gatecompletecourse

1. Quality Learning: No Rote-Learning. Understand Everything, from basics, In-depth, with variations.
2. Daily Homeworks, Quality Practice Sets, Weekly Quizzes.
3. Summary Lectures for Quick Revision.
4. Detailed Video Solutions of Previous ALL GATE Questions.
5. Doubt Resolution, Revision, Practice, a lot more.

Download the GO Classes Android App:

https://play.google.com/store/apps/details?id=com.goclasses.courses

Search “GO Classes” on Play Store.



NOTE :

Complete Discrete Mathematics & Complete Engineering Mathematics Courses, by GO Classes, are **FREE** for ALL learners.

Visit here to watch : https://www.goclasses.in/s/store/

SignUp/Login on Goclasses website for free and start learning.

We are on Telegram. **Contact us** for any help.

Link in the Description!!

Join GO Classes **Doubt Discussion** Telegram Group :

Username:

@GATECSE_Goclasses

We are on Telegram. **Contact us** for any help.

Join GO Classes Telegram Channel, Username: @GOCLASSES_CSE

Join GO Classes **Doubt Discussion** Telegram Group :

Username: @GATECSE_Goclasses

(Any doubt related to Goclasses Courses can also be asked here.)

Join GATEOverflow Doubt Discussion Telegram Group :

Username: @GateOverflow_CSE

Next Chapter:

First Order Logic

Propositional logic

Propositions

True

false

FOL

objects

& their properties

FOL =

1. prop. logic

extra features of FOL

2. objects (Domain)
3. their properties (Predicate)
4. multiple objects properties (Quantifier)

First Order Logic:

1. Objects: Domain
2. Properties of Objects: Predicates
3. Talk about Multiple Objects: Quantifiers

First Order Logic

Next Topic:

Objects, Domain

Prop. logic:

variable

P $\rightarrow$ T, F

Propositional Var.
(Boolean Var.)

Domain (Universe) $\rightarrow$ Set of values a variable can take

The Universe of Propositional Logic

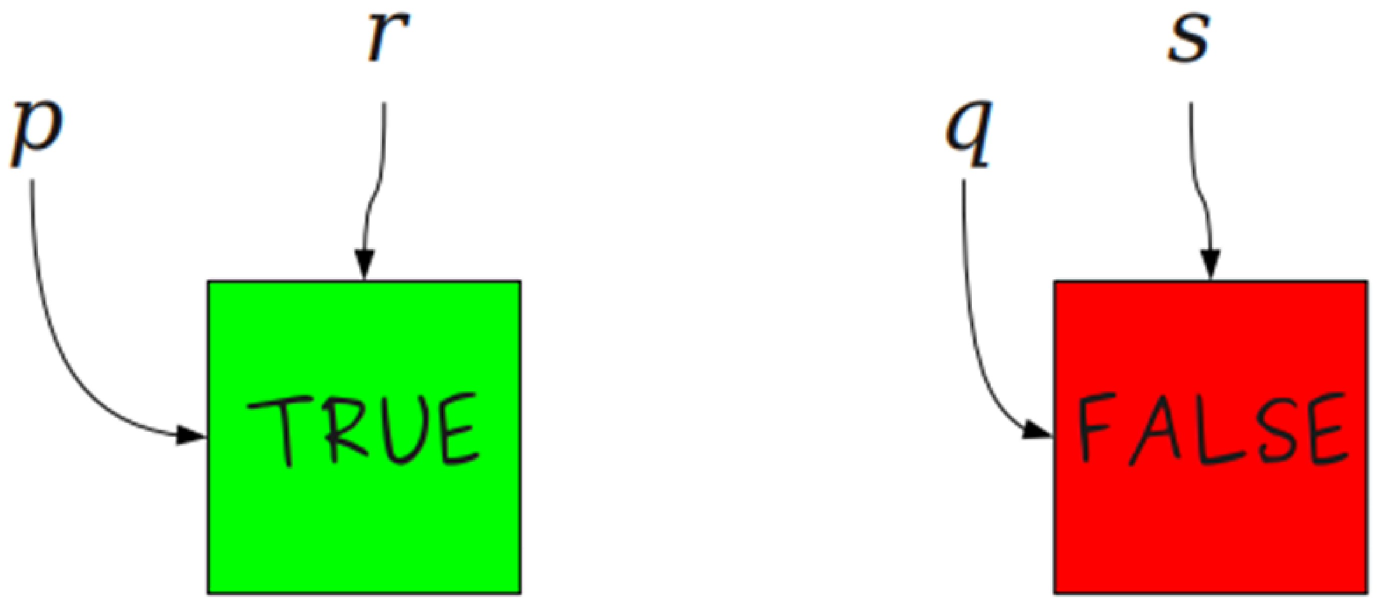
$$p \wedge q \rightarrow \neg r \vee \neg s$$

TRUE

FALSE

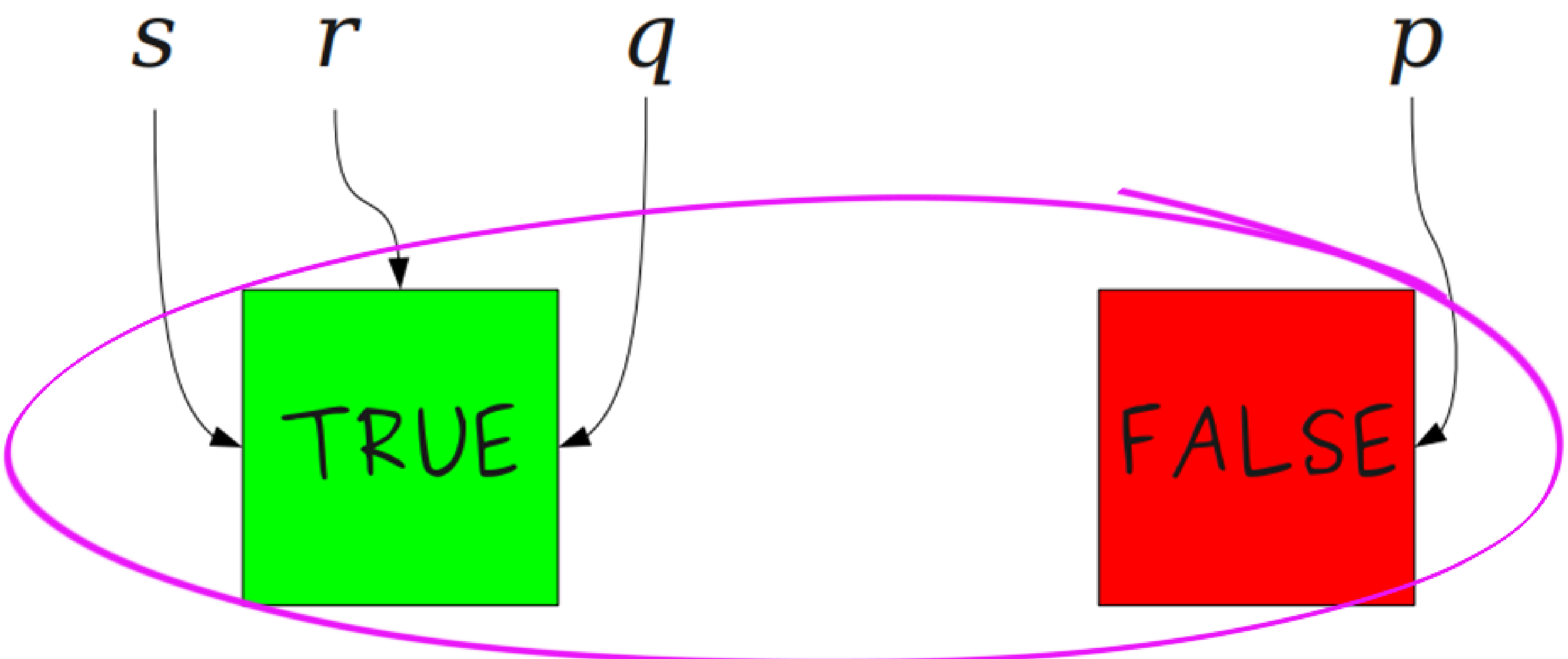
The Universe of Propositional Logic

$$\boxed{p \wedge q \rightarrow \neg r \vee \neg s} = \text{True}$$



The Universe of Propositional Logic

$$\boxed{\overset{F}{p \wedge \underline{q}} \rightarrow \neg r \vee \neg s} = \text{True}$$



Propositional Logic

- In propositional logic, each variable represents a **proposition**, which is either true or false.
- Consequently, we can directly apply connectives to propositions:
  - $p \rightarrow q$
  - $\neg p \wedge q$
- The truth or falsity of a statement can be determined by plugging in the truth values for the input propositions and computing the result.
- We can see all possible truth values for a statement by checking all possible truth assignments to its variables.

Propositional Logic:

A World of True, False.

A world of Facts(statements/propositions).

First Order Logic:

Just like a Real World(People & their Stories)

A World of Objects & their properties.

Domain

Predicate

Propositional Logic:

A World of True, False.

A world of Facts(statements/propositions).

First Order Logic:

A World of Objects & their properties.

First Order Logic:

1. Objects: Domain

2. Properties of Objects: Predicates

3. Talk about Multiple Objects: Quantifiers

Domain or Universe:

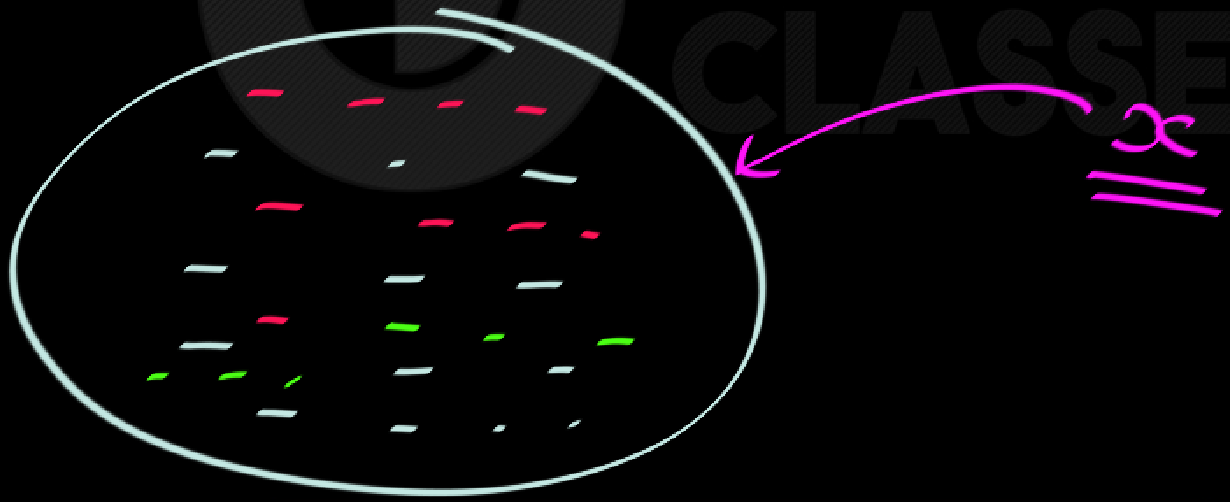
In FOL, we talk about Objects.

The Set of Objects that we discuss is called the Domain or Universe.

i.e. The Set of Objects of interest is called the Domain or Universe.

Objects of intereset : People

Domain : Set of people



Objects of intereset : Real numbers

Domain : Set of Real Numbers

Domain

0, 0.11, π, −1.33, ...

x

Domain: Set of Values a Variable can take.

Objects of intereset : Natural Numbers

Domain : Set of all Natural Numbers

1, 2, 3, 4, — — — — — — — — — —

object (Constant)

x

y

$x \neq 0.5$

$y \neq$ Car

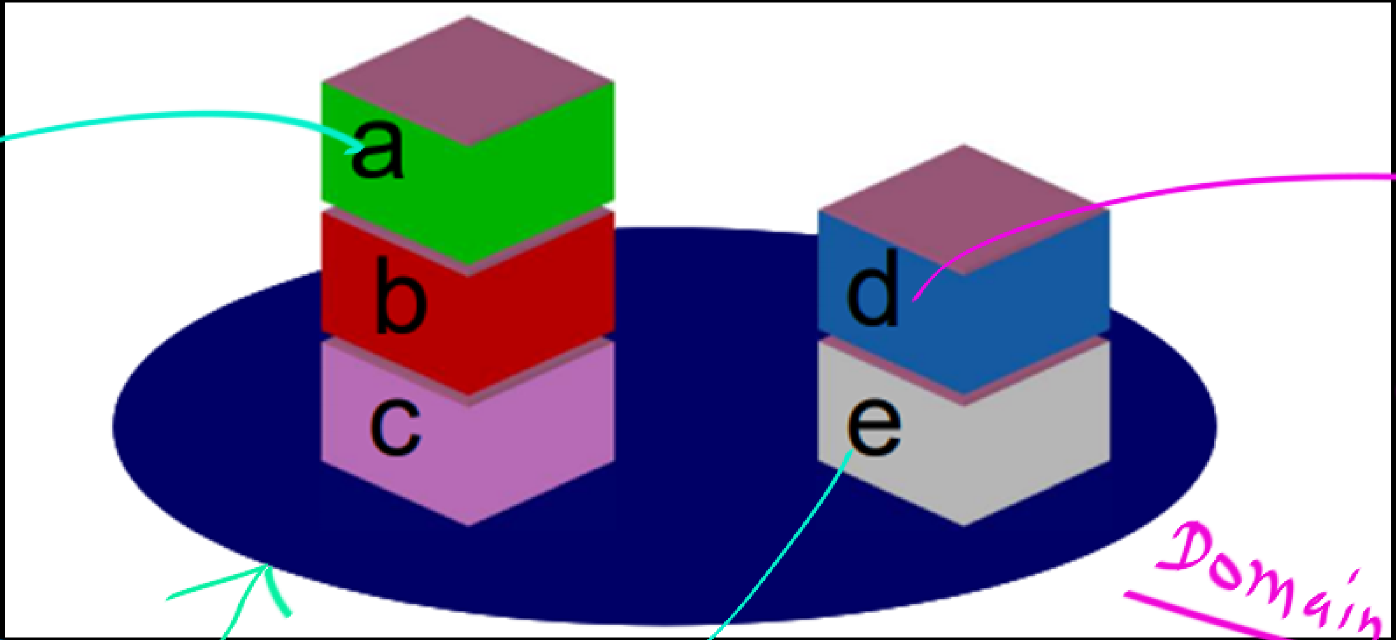
Objects of intereset :

Domain : : Set of objects of interest

Set of values a variable can take.

a, b, c,
_ _ _
_ _ _
_ _

Variable x



object

object (Constant)

Domain = {a, b, c, d, e}

Variable
x

object

First-Order Logic speaks about *objects*, which are the domain of discourse or the universe. First-Order Logic is also concerned about *Properties* of these objects (called *Predicates*), and the *Names* of these objects.

So we begin by fixing a domain of interest, called the

domain of discourse: $\mathcal{D}$.

This can be any ***nonempty*** set at all – e.g., a set of (actual or imaginary) blocks on a table that you want to encode knowledge about; a set of numbers or other mathematical objects; a set of possible electronic circuits and their components; the set of all physical objects in the universe; the set of all physical objects, events, cultural and social entities in the world; etc.

First-Order Logic speaks about objects, which are the domain of discourse or the universe.

First-Order Logic

- In first-order logic, each variable refers to some object in a set called the **domain of discourse**.

FOL

Variable

x

Domain

Prop. logic:

Variable

p

T, F

Domain

Notation

- Variables: $x, y, z, \ldots$
  - Represent elements of an underlying set
- Constants: $a, b, c, \ldots$
  - Specific elements of underlying set

Domain

Domain

Constant ≡ object in the Domain

Prop. Logic vs. FOL

- Propositional logic assumes that there are facts that either hold or do not hold
- FOL assumes more: the world consists of objects with certain relations among them that do or do not hold

First-Order Logic (FOL)

- Whereas propositional logic assumes the world contains facts

- First-order logic (like natural language) assumes the world contains
  - Objects: people, houses, numbers, colors, baseball games, wars, …
  - Relations: red, round, prime, brother of, bigger than, part of, comes between, …

What is First-Order Logic?

- ***First-order logic*** is a logical system for reasoning about properties of objects.
- Augments the logical connectives from propositional logic with
  - ***predicates*** that describe properties of objects,
  - ***functions*** that map objects to one another, and
  - ***quantifiers*** that allow us to reason about multiple objects.

First Order Logic

Next Topic:

Predicates

(Properties of Objects)

First Order Logic:

Just like a Real World(People & their Stories)

A World of Objects & their properties.

Domain

predicate

Predicate:

Predicate tells us about the Properties of Objects.. and Relationship among objects.

Example 1: Domain: Set of people

Variable

male(x) : $\boxed{x \text{ is a male.}}$ ✓

male(John) : True

male(Sonu) : false

Sonu is a male : false

D

Sonu, Ram, John, ...

Variable x

Example 1: Domain: Set of people

Consider the following predicates defined over the set of people:

- F(x) = "x is a friend of mine"
- C(x) = "x is clever"
- W(x) = "x is wealthy"
- B(x) = "x is boring"

predicate

Property of objects in the Domain

predicate

Domain: people

Variable

x

Predicate: tells us about:

Properties of objects

& Relationship among objects

Example 1: Domain: Set of people

$f(x, y)$: x is father of y.

$f(a, b)$: True

$f(b, a)$: false

$f(a, a)$: false

Assume a is father of b.

Domain: people

a, b, c, d, ...

Example 1: Domain: Set of people

x

y

Predicates:

$S(x)$: x is a student. → Unary predicate

$F(x, y)$: x and y are friends. → binary predicate

$O(x, y)$: x is older than y.

S(x) : x is a student.

Unary predicate : Predicate over one Variable

Example 1: Domain: Set of people

$C(x,y,z)$: x, y, z are classmates.

ternary predicate : predicate over three variables.

Predicate:

Domain: D

$P(x_1, x_2, \ldots, x_n)$: n-ary predicate

variable

variable

Example 2: Domain: Set of Numbers

Real

Positive(x) : x is +ve

Positive(-0.5) = False

Positive(π) : True

Positive(0) = false

Variable
x

y

D

$0, 1, 10, \pi, -0.5, \ldots, -9.212, \ldots$

Example 2: Domain: Set of Numbers

$GE(x, y)$: $x \geq y$

binary predicate

Domain : Real No.

x y

$GE(4, 2)$: True

$GE(2, 4)$: false

$GE(2, 2)$: True

Example 2: Domain: Set of Numbers

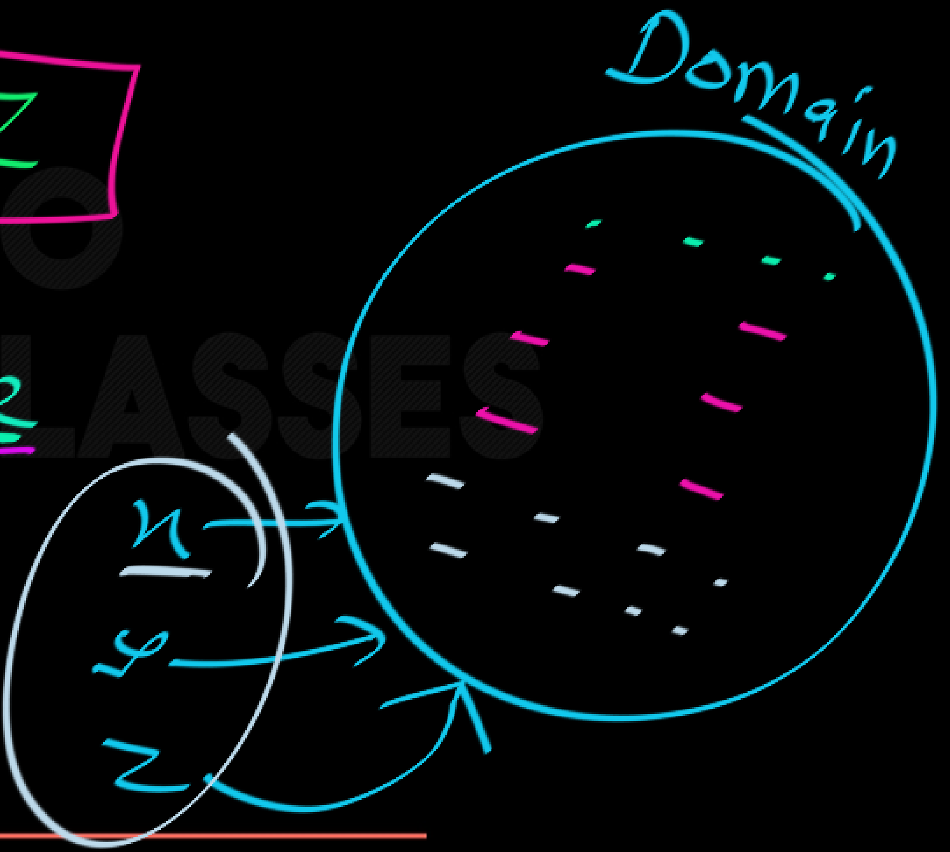
$P(x, y, z)$: $\boxed{x + y = z}$

Ternary Predicate

$P(0,0,0)$: True

$P(5,2,3)$: false

$P(2,3,5)$: True



Predicate :

n-ary predicate

$P(x_1, x_2, ---, x_n)$

Variable

Domain

$P(a_1, a_2, --, a_n)$: Proposition

True/false

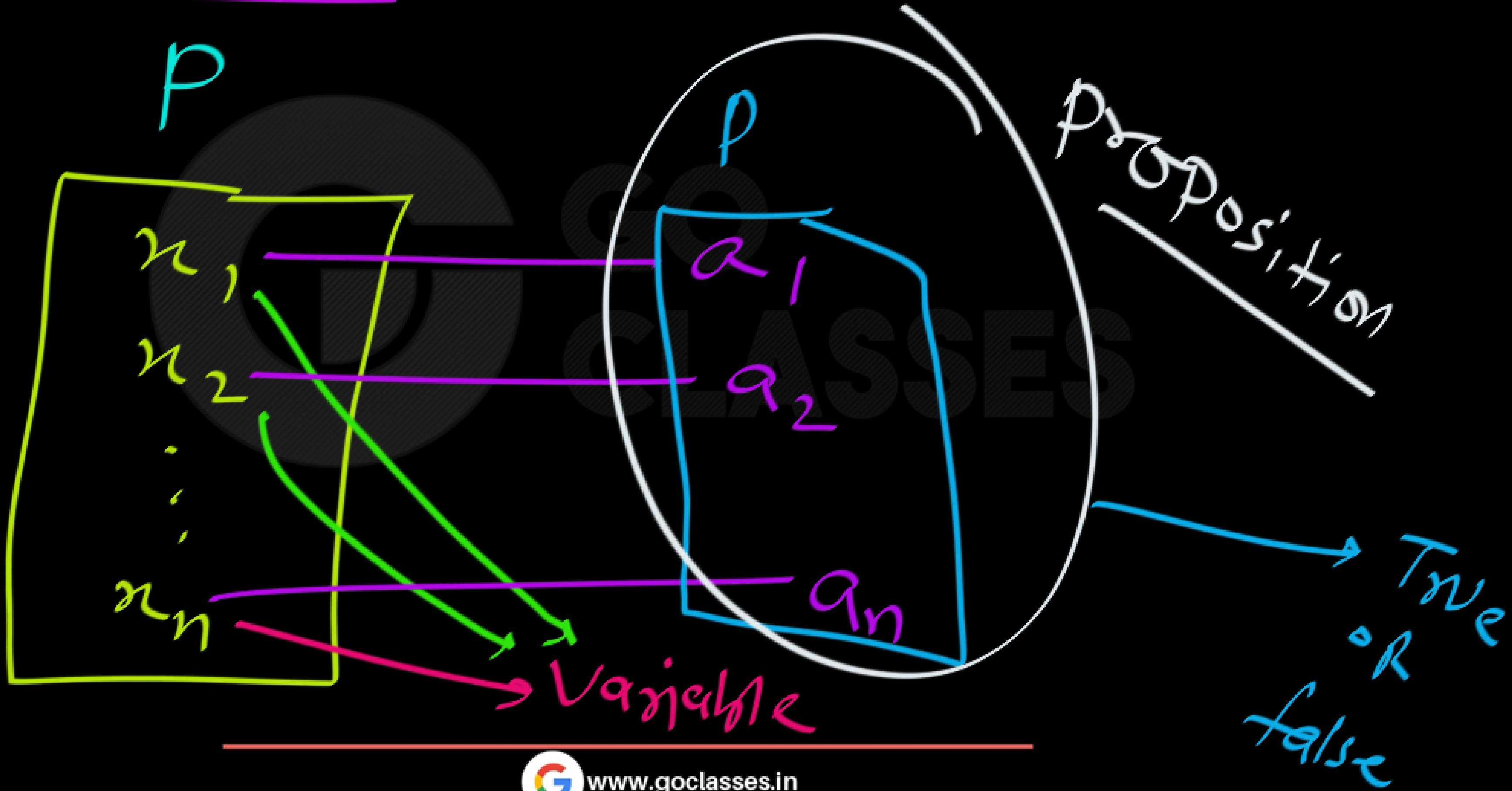
Predicate:

a sentance containing variables

(every variable refers to the Domain)

such that it becomes a proposition once we Replace each variable with specific value from Domain.

Predicate



What is a predicate?

A "predicate" is a statement involving variables over a specified "domain" (set). Domain is **understood ahead of time**

Example

Domain (set)	Predicate
Integers ($\mathbb{Z}$)	$S(x)$: x is a perfect square
Reals ($\mathbb{R}$)	$G(x, y)$: $x > y$
Computers	$A(c)$: c is under attack
Computers; People	$B(c, p)$: c is under attack by p

Domain: $\mathbb{Z}$

$S(x)$: $\boxed{x \text{ is a perfect square.}}$

$S(9)$: True

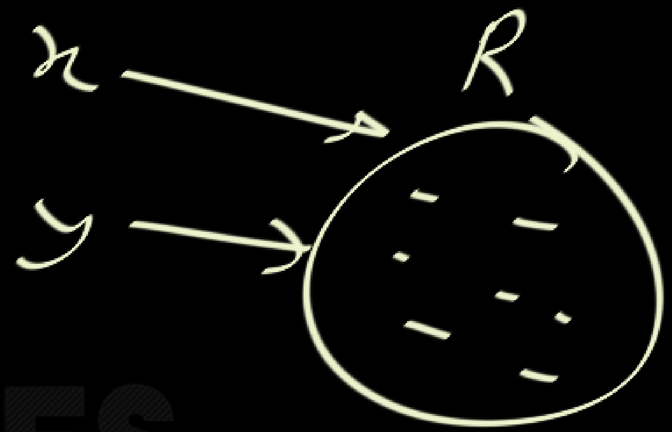
$S(7)$: false

$S(20)$ Proposition

false

Domain: $\mathbb{R}$

$G(x,y)$: $x > y$



$G(4,7)$: false

$G(7,4)$: True

$G(0,-10)$: Proposition

True

Domain :

Computers + people

$c_1, c_2, \ldots$

John, Sonu ...

$B(c, p)$: $\boxed{\text{Computer } c \text{ is under Attack by Person } p.}$

Predicate:

In simple words, **Sentence involving Variables.**

Consider the following statements:

$$x > 3, \quad x = y + 3, \quad x + y = z$$

The truth value of these statements has no meaning without specifying the values of x, y, z.

Unary Predicate

binary Predicate

ternary Predicate

Predicate

Let U be some universal set.

A logical expression containing some *variable* that becomes a proposition when we *substitute* any particular value from the universe for this variable is called a **predicate**.

It is also called *propositional function*.

Predicate:

A statement involving variables, where every variable has a Domain/universe, and when we replace ALL variable with specific values from their respective Domain, we get a proposition.

Different variables may have diff domains.

Example

- Let P(x, y) = "x > y".
  Domain: integers, i.e. both x and y are integers.
- P(4, 3) means "4 >3", so P(4, 3) is TRUE;

  P(2,2) : False
- P(1, 2) means "1>2" , so P(1, 2) is FALSE;

  P(2,1) : True
- P(3,4) is false (in general, P(x,y) and P(y,x) not equal).

Propositional Functions: Example

Domain: $\boxed{\text{Natural Numbers}}$

- Let $Q(x,y,z)$ denote the statement '$x^2+y^2=z^2$'
  - What is the truth value of $Q(3,4,5)$?

    Q(3,4,5) is true
  - What is the truth value of $Q(2,2,3)$?

    false

    $\boxed{2^2+2^2=3^2}$

    Q(2,3,3) is false

An example

Domain: N

Example:
Let's consider the following expression:

$\boxed{\textit{Natural number } x \textit{ is even.}}$ → predicate

Is it a proposition? No
Is it a kind of logical statement?
What can make it a proposition?

Substituting any particular value for the variable x (e.g. 3) will make it a proposition, e.g.:

Natural number 2 is even. True

Natural number 3 is even. False

Domain : $\mathbb{Z}$

Predicates

- Is the statement "x^2 is greater than x" a proposition?

- Define P(x) = $\boxed{\text{'}x^2 \text{ is greater than } x\text{'.}}$
- Is P(1) a proposition? P(1)="1^2 is greater than 1" (F)

P(2) : True ; P(-5) : True

A predicate is a statement that contains variables (predicate variables) and that may be true or false depending on the values of these variables.

Predicates take objects as arguments and evaluate to true or false.

elements of Domain

Hence, predicates are also called "propositional function" Because predicates become proposition when we replace every variable by objects from domain of that variable.

$\boxed{\text{Predicate} \equiv \text{Propositional function}}$

Note 1:

Note: Domain in FoL is ALWAYS Non-empty; unless explicitely stated.

$P(x,y)$: x has Salary y.

Domain

John, – –

2000, – – –

$P(x, y)$; x has salary y.

People

Numbers

NOTE 2:

By default (if nothing is mentioned about domain, or domain of different variables) Domain of every variable in a FOL expression is same, and it is called "Domain/universe of the expression"

Since a predicate takes value true or false once instantiated (that is, once its variables are taking values), we may alternatively say that a predicate instantiated becomes a proposition.

Predicate Instantiated/Domain

A predicate is a statement that contains variables (predicate variables) and that may be true or false depending on the values of these variables.

- A predicate instantiated (where variables are evaluated in specific values) is a proposition.

The domain of a predicate variable is the collection of all possible values that the variable may take.

  - e.g. the domain of x in P(x): integer
  - Different variables may have different domains.

- Predicate logic extends (is more powerful than) propositional logic.

First Order Logic

Next Topic:

Making Proposition from Predicate

Q :

How to make "Propositions" from predicates?

Consider the following statements:

Domain: Z

$$x > 3, \quad x = y + 3, \quad x + y = z$$

The truth value of these statements has no meaning without specifying the values of x, y, z.

However, we *can* make propositions out of such statements.

Domain: $\mathbb{Z}$

$P(x, y)$: $\boxed{x = y + 3}$

$P(-5, 0)$ Proposition

$P(-5, 0)$: $\boxed{-5 = 0 + 3}$ false

Q :

How to make "Propositions" from predicates?

Ans:

Way 1 : Substitute every variable by some value from the domain.

Q :

How to make “Propositions” from predicates?

Ans:

Way 1 : Substitute every variable by some value from the domain.

Is there Any Other Way??

Domain: $\mathbb{Z}$

$E(x)$: $\boxed{x \text{ is even.}}$ Predicate

$E(19)$: false

proposition

Domain: $\mathbb{Z}$

$E(x)$: x is even.

for all x in the Domain, $E(x)$ is True. → proposition → false

$\equiv$ every integer is even : proposition → false

Domain: $\mathbb{Z}$

$E(x)$: $\boxed{x \text{ is even.}}$

there exists x in the Domain, $E(x)$ is True.

$\equiv$ $\boxed{\text{At least one integer is even.}}$ Proposition

→ True

for all ----

there exists ----

Can create Propositions

Quantification words

Q :

How to make "Propositions" from predicates?

Ans:

Way 1 : Substitute every variable by some value from the domain.

Way 2 : Quantification

Quantifiers

Introduction

A predicate becomes a proposition when we assign it fixed values. However, another way to make a predicate into a proposition is to *quantify* it. That is, the predicate is true (or false) for *all* possible values in the universe of discourse or for *some* value(s) in the universe of discourse.

Such *quantification* can be done with two *quantifiers*: the *universal* quantifier and the *existential* quantifier.

Quantification

- Statements like
  - Some birds are angry.
  - On the internet, no one knows who you are.
  - The square of any real number is nonnegative.

Quantification words in English:

few, All, many, some, any ...

Quantification words in logic:

All $\equiv$ every

"Some" $\equiv$ "at least one"

In FOL:

→ two Quantifiers:

(1) for all ≡ every ≡ universal quantifier

(2) there exists ≡ at least one

≡ existential quantifier

Quantifiers

Standard

- Universal Quantifier :
  talks about "Universal" Quantification
  → over all elements in Domain
- Existential Quantifier
  talks about "Existence"
  → Does there exist some element.

Quantifier:

→ a way of creating proposition from predicate.

Domain: $\boxed{\text{set of all people.}}$

male(x): $\boxed{\text{x is a male.}}$ → Predicate

male(John): True ;

male(Sonu): false

→ Proposition

Domain: $\boxed{\text{set of all people.}}$

male(x): $\boxed{\text{x is a male.}}$ → Predicate

$\boxed{\text{for all x in the Domain, male(x) is True.}}$

$\equiv$ $\boxed{\text{Every person is a male}}$: false

proposition

Domain: $\boxed{\text{set of all people.}}$

male(x): $\boxed{x \text{ is a male.}}$ → Predicate

$\boxed{\text{There exists } x \text{ in the Domain, male}(x) \text{ is True}}$

$\equiv$ $\boxed{\text{at least one person is male}}$: True

Proposition

First Order Logic

Next Topic:

Universal Quantifier

(for all)