Sets, Relations & Functions

Page No.
Date

Set (S):

collection of unordered, singular elements seperated by comma.

For any element e
$e \in S$: e belongs to set S
$e \notin S$ e does not belong to set S.

Predicate method: condition imposed on variable

equivalent
$i < 5 \equiv P(i)$

name of predication
name given to condition

Using substitution, i can be replaced by some concrete variable. And then, we can evaluate whether predicate holds true for given variable.

Values that can replace variable are called $\boxed{\text{domain of predicate.}}$

$P(i)$: $6 \leq i \leq 18$
$P(K)$: $K \% 2 == 0$
$P(l)$: l is natural no
$l \in N$ (l belongs to set no natural nos)

P of l such that l is...

Set has power to form a collection using predicate variable, domain of predicate & predicate (condition)

$$S = \{ n \in N \mid 1 \le n \le 6 \}$$

predicate

set of natural numbers 1 to 6

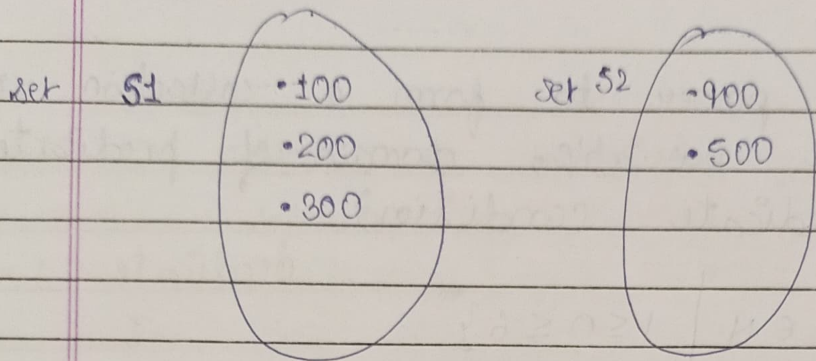
predicate variable

domain of predicate

Name Set = {PredicateVar ∈ Domain of predicate : predicate}

→ '|' and ':' stands for 'such that' in mathematics.

Cartesian products of sets



Cartesian product of sets $= S1 \times S2$

$S1 \times S2 = \{(x, y) \mid x \in S1 \text{ and } y \in S2\}$

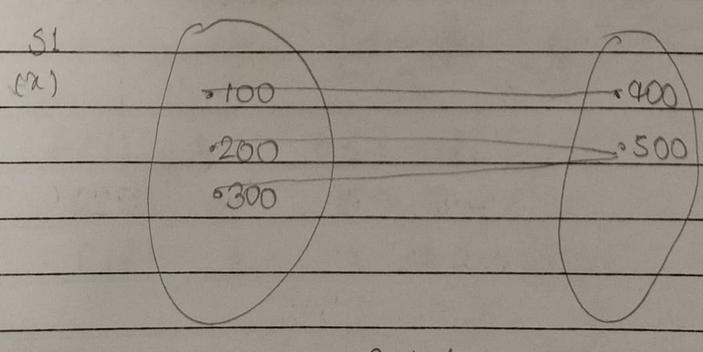
$= \{ (100, 400), (100, 500)$
$(200, 400), (200, 500)$
$(300, 400), (300, 500)$
$\}$

Relation is non-empty subset of $S1 \times S2$

$R: S1 \to S2 \subseteq S1 \times S2$

$R: S1 \to S2 \underset{\text{is not}}{\neq} \phi$ (empty)

$\{ (100, 400), (300, 500), (200, 500) \}$



Relation

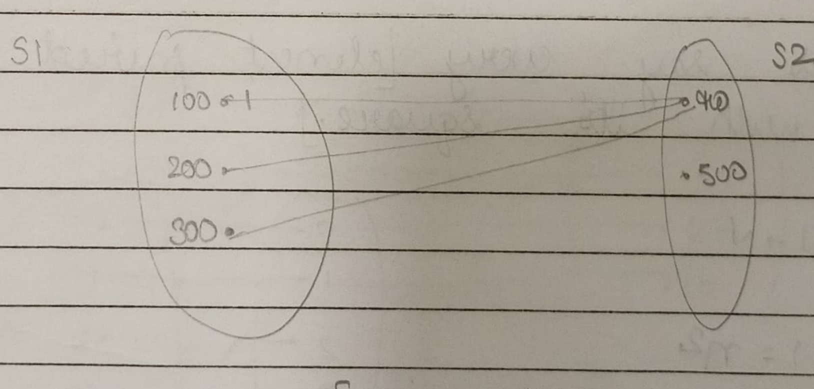
Function bhi relation hi hai but with some rule.

Function: A function is relation $(\subseteq S1 \times S2)$ adhereing to following rules

Rule: Every element in set S1 must be paired with exactly one element in S2.

ALWAYS TRUE? → yes 1 output must be present for any input

Let S1, S2 be two non empty sets. Any non - empty subset of $S1 \times S2$ which is formed by selecting ordered pairs so that [every element in S1 forms a pair with exactly one element in S2] is known as function from S1 to S2
$f: S1 \rightarrow S2$

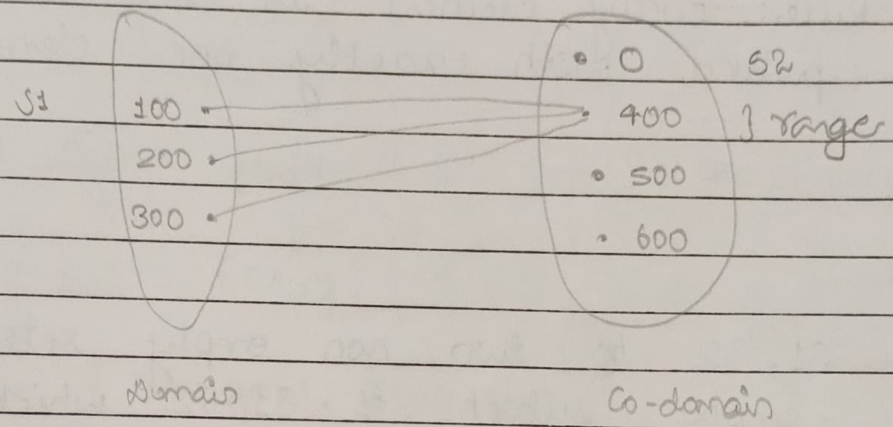


$f: S1 \rightarrow S2$ (many - one)

Every element in S1 forms relationship with exactly one element in S2

Range of function : $R_f \subseteq$ Codomain (S2)

R_f : { y | y is paired with at least one element from domain }

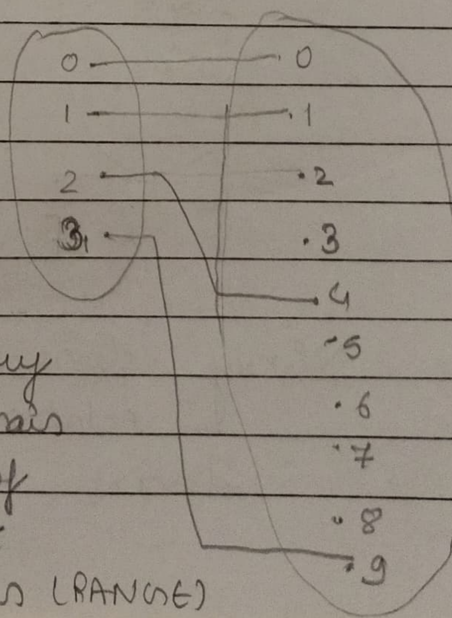


In real world, every element in domain is paired with unique element in co-domain (f: N → N)

- Lets say every [element paired with it's square.]

f: N → N

$f(n) = n^2$



f(n): name of rule

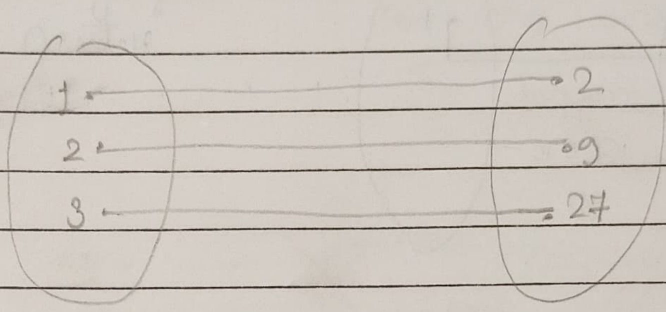
n: representation of every element in domain

n2: representation of paired element in codomain (RANGE)

```
def (n):
    return n×n
```
} representation in Python

1. Let's say a function from natural numbers to natural numbers, $f: N \to N$.

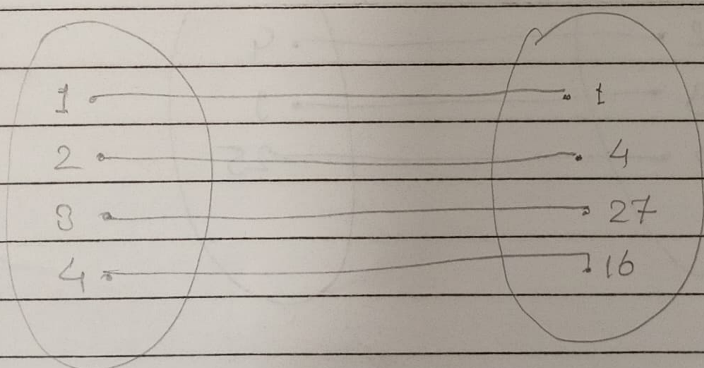
Every element is mapped to it's cube plus one.



name of rule

$f(n) = \underbrace{n^3 + 1}$ representation of paired element in co-domain

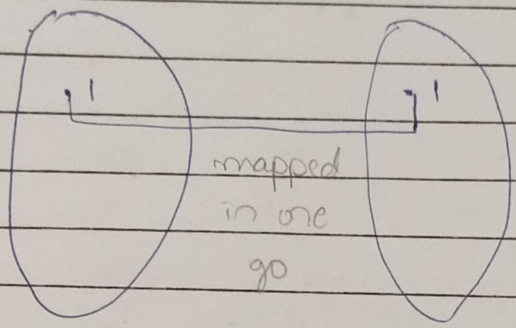
2. Let's try to map even elements to square & odd elements to cube.



$f(n) = n^2$ (n is even)
$\quad\;\; = n^3$ (n is odd)

1. Let's map every no to its square
$f(n) = n^2$ (for all n)

$f(1) =$

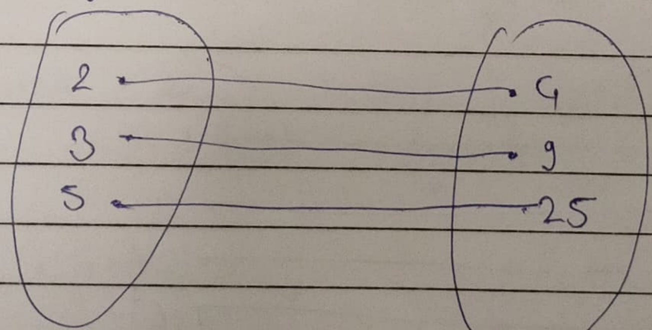


code
```
def f(n):
    return n×n
```

```
void f (int n) {
    return n * n;
}
```

Works for $n \in \mathbb{N}$

1. $f(n) = n^2$



→ Mapped in one go
for every input

Mathematically kar ke dry run.

MAIN

Trying to find out factorial recursively.

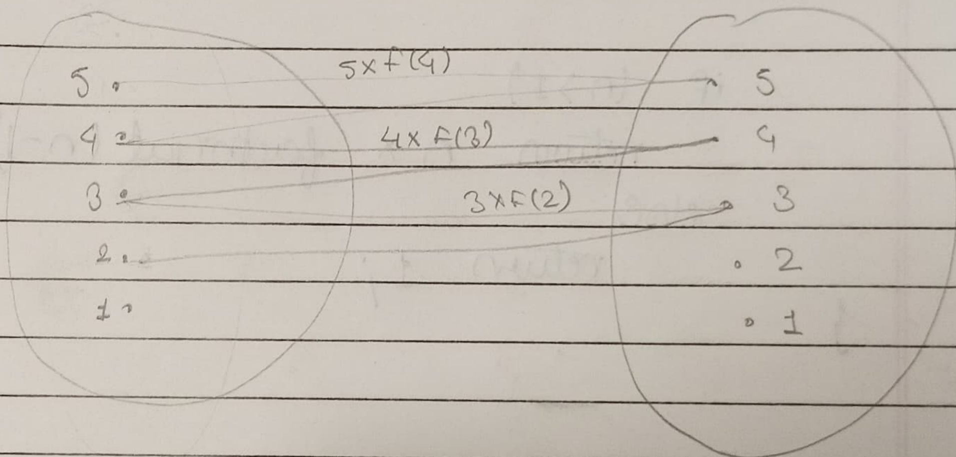
$f(5) = 5 \times 4!$

MATHS $f(n) = n * f(n-1)$ $n \geq 2$

$= 1$ $0 \leq n \leq 1$

For my understanding

$f(5) =$



Mathematically $5! = 5 \times 4!$

$\because$ $factorial(n) = \begin{cases} n \times factorial(n-1) & n \geq 2 \\ 1 & n = 1 \end{cases}$

```
long long unsigned int (factorial (int n) {

    if (n>1)
        return n x factorial (n-1);
    else
        return 1;
}
```

Little DS/ code
then maths
then actual code (dry run)