

# List in Prolog - :

✓ Lists of multiple elements.

✓ All these elements can be atoms.

• Elements inside the list are separated by comma.

List is denoted by [ ].

Integer 1, 2, 3, 4,  
12, 4

alphabets & numbers.

{ 12, 13

{ 10, 8, 6

$x = y ; x = 2$

grocery\_list = [apple, sugar, rice, banana]

{ 12, 13

[apple, sugar, rice, banana]

Head, tail notation  $[X | Y]$

first element of the list is the head of the list

$Y = \text{sugar, rice, banana}$

tail = [sugar, rice, banana]  
[apple | tail]

grocery\_list ([apple, sugar, rice, banana]).

① know the number of  
elements

?- grocery\_list ([w, x, y, z]).

?- grocery\_list ([X]).

Swi-Prolog  
switch

→ ;  
→ ;  
→ ;

next

② Don't know the number of  
elements.

neeraj, cexmc3  
arcey, cex422  
neeraj, I1912K  
neeraj,

$[1, [2, 3]]$  Head = 1, tail =  $[2, 3]$

$[[1, 2], 3]$  Head =  $[1, 2]$ , tail = 3

---

Mapping in lists =

$\text{names\_list}([x, y, z]) = \text{names\_list}(\text{[vidyut, shikha, atul]})$

Relationship  
function

$x = \text{vidyut}$

$y = \text{shikha}$

$z = \text{atul}$

Equality & unification

Recursion on stack

# Equality v/s Unification -

$$X = Y$$

neither equality, nor assignment  
unification

? -  $X = Y$  whether variables  $X$  &  $Y$  [mapping]

Rules for unification -

radius ( $x, 3$ )

$\Rightarrow$

can be unified or not

Rule 1 -  $X = Y$

Equality/unification among variables.

• Uninstantiated variables - if should be empty.

teaches(neeraj, cer4c3).

$Y = ?$

? - teaches(neeraj,  $X$ ).

$X \neq \text{cer4c3}$   
 $X = \text{cer4c3}$

teaches(lalit, cer4c2).

? - teaches(lalit,  $Y$ )  
 $Y = \text{cer4c2}$

$X = Y$   $\hookrightarrow$  false

## Rule 2

Unification among integers and atoms

$$?- \text{cer}(4, 3) = \text{cer}(4, 3)$$

↓  
↑  
true atom

↑  
atom

$$\text{cer}(4, 3) = 10$$

↑ false integer  
atom

$$?- \text{cer}(4, 2) = \text{cer}(4, 3)$$

↑ atoms      ↑ atoms

false

$$?- 784 = 784 = \text{True}$$

↑ integer      ↑ integer

$$?- 784 = 26 = \text{false}$$

Prolog 2022 AI  
Rule-based systems  
System design  
soft computing  
Alan Turing  
Turing

Turing Test  
chatbots [1950s]

⇒ Question (How's the weather, it's a sunny day)

? - Question (How's the weather, X)

↪

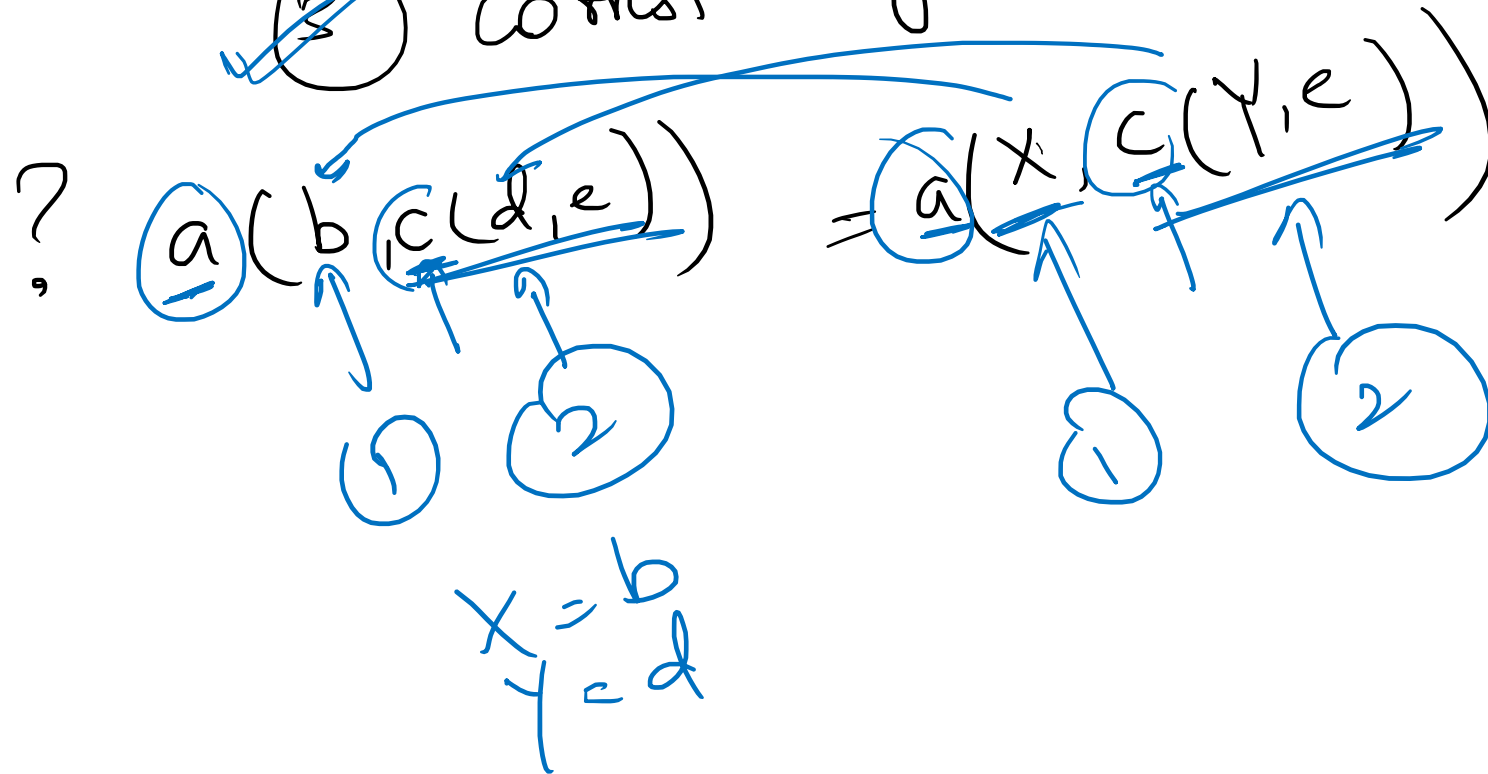
X = It's a sunny day.

Question (X, Y)  
Question (X, Y)  
Question (X, Y)  
Question (X, Y)

# Rule 3 - verification among structures -

Structures can be declared unifiable -

- 1 They have the same functors.
- 2 They have the same number of components.
- 3 Corresponding components should also be equal.



unifiable  
mapping



$\text{plays}(\text{matt}, \text{Character}(\text{joey}, \text{friends})) = \text{plays}(X, \text{Character}(Y, \text{friends}))$

$X = \text{matt}$

$Y = \underline{\text{joey}}$

?-  $\text{anime}(A, \text{tyson}) = \text{anime}(\text{max}, B)$ .

$A = \text{max}, B = \text{tyson}$

?-  $\text{rivers}(X, Y, Z) = \text{rivers}(\text{ganga}, \text{yamuna}, \text{narmada})$ .

?-  $\underline{\text{letter}}(X) = \underline{\text{word}}(\text{letter})$

-- false

structure  
let(s) = 18 atom  
false

? -  $f(x, x) = f(10, 20)$ .

false

overwrite prod

Equality in variable

- uninstantiated
- integers & atoms are always equal to one another
-

