

Pre-LAB:-

(ii) We need to Unify $p(x, y, y)$ with $p(a, z, b)$.

Unification :-

Unification is a process of making two different logical atomic expressions identical by finding a substitution.

Unification depends on the substitution process.

It takes 2 literals as input and makes them identical using substitution.

Let ψ_1 and ψ_2 be two atomic sentences and α be a unifier such that $\psi_1^\alpha = \psi_2^\alpha$ then it can be expressed as $\text{UNIFY}(\psi_1, \psi_2)$.

then it can be expressed as

UNIFY (φ_1, φ_2)

No, let $\varphi_1 = \text{king}(x)$, $\varphi_2 = \text{king}(\text{john})$

substitution $\Theta = \{\text{john}/x\}$ is a unifier
for these atoms and applying
this substitution, and both expression
will be identical.

We have some conditions for
Unification :- like

- * Predicate symbol must be same,
atoms or expression with different
predicate symbol can never be
unified.
- * Number of arguments in both
expressions must be identical.
- * Unification will fail if there
are 2 similar variables present
in the same expression.

INLAB:-

We are a problem where:-

A,B,C,D & E are 5 thieves living on different floors of a 5 storied apartment. They escaped from the police and are hiding in same building.

The problem statements are:-

- * A does not live on top floor.
- * B does not live on bottom floor.
- * C does not live on either top or bottom floor.
- * D lives on higher floor than B.
- * E does not live on a floor adjacent to C.
- * C does not live on a floor adjacent to B.

We need to write the python code, to help the policemen to find out where do the thieves live, so let's implement the problem using the given problem statements. and the output comes in such a way that, we will get the order in which thieves stay in the apartment.

POST LAB:-

Ques. We have 2 expressions $p(x)$ and $p(y)$. One way to unify these is to substitute any constant expression for x and y : $\{ \text{fred}/x, \text{fred}/y \}$. But this is not the most general unifier, because if we substitute any variable for x & y we get a more general unifier.

$\{z/x, z/y\}$. The first Unifier
is a valid unifier, but it would lessen
the generality of inferences that
we might want to make.

190030117-AILAB_9_INLAB:OUTPUT:

The screenshot shows a Jupyter Notebook interface with three tabs at the top: LAB_10_IN_1 - Colaboratory, LAB_10_IN_2 - Colaboratory, and LAB_9_INLAB - Colaboratory. The LAB_9_INLAB tab is active. The notebook contains the following Python code:

```
[1] import random
[2] floors={1,2,3,4,5}
    a_lives_in=[1,2,3,4]
    b_lives_in=[2,3,4,5]
    c_lives_in=[2,3,4]
[3] check=True
[4] while check:
    b=random.choice(tuple(b_lives_in))
    if b==2:
        c=b+2
        if c in c_lives_in:
            check=False
    e=c-1
    floors_occ=(b,c,e)
    a=a_lives_in-floors_occ
    floors_occ.update(a)
    d=floors-floors_occ
```

A tooltip is visible over the code in step 4, showing the variable definitions: `e=c-1`, `floors_occ=(b,c,e)`, `a=a_lives_in-floors_occ`, `floors_occ.update(a)`, and `d=floors-floors_occ`. The status bar at the bottom indicates "Saving...".