Name : S . SANJAY
Roll No : 241801246
Dept : AI & DS- FD
Date : 17 . 05. 2025

6 : Implementation of Unification and Resolution

Algorithm

```
#Function to unify two expressions
 def unify(x, y, theta=None):
if theta is None:
theta={} if x == y:
      return theta
elif is instance(x,str)and x.islower():  #variable
return unify_var(x, y, theta)
      elif is instance(y,str)andy.islower():#variable
        return unify_var(y, x, theta)
elif is instance(x,list)andisinstance(y,list)andlen(x)==len(y):
for xi, yi in zip(x, y):
        theta=unify(xi,yi,theta) if
        theta is None:
          returnNone
      return theta
   else:
      returnNone
```

```
  defunify_var(var,x,theta):
    if var in theta:
    return unify(theta[var],x,theta) elif
  x in theta:
    return unify(var,theta[x],theta)
  else:
    theta[var]=x return
    theta

#Resolution- like function for simple implication

def resolution(kb, query):

for clause in kb:
premise,conclusion=clause
theta=unify(conclusion,query,{}) if theta is not None:

#Check if all premises unify

        all_premises_true=all(unify(p,fact,theta)isnotNoneforfactinfactsforpin[premise]) if
        all_premises_true:
          returnTrue
  return False


#Knowledgebase:Implication-Human(John)→Mortal(John)
knowledge_base = [
  [["Human","x"],["Mortal","x"]]    #generalized implication
]


#Known facts

facts=[["Human","John"]]
```

```
# Query
query=["Mortal","John"]


#Check resolution
  if resolution(knowledge_base, query):
    print("Query is resolved: John is Mortal")
else:
  print("Query could not be resolved")
```