**5.Implementation of Backward Chaining**

**Aim:** To Implementation of Backward Chaining

**Code:**

knowledge\_base = {

'mammal': [['has\_hair']],

'mammal': [['gives\_milk']],

'bird': [['has\_feathers']],

'carnivore': [['eats\_meat']],

'carnivore': [['has\_pointed\_teeth', 'has\_claws', 'has\_forward\_eyes']],

'ungulate': [['mammal', 'has\_hooves']],

'cheetah': [['mammal', 'carnivore', 'has\_spotted\_coat']],

'zebra': [['mammal', 'ungulate', 'has\_black\_stripes']],

'ostrich': [['bird', 'is\_long\_necked']],

'penguin': [['bird', 'swims']],

'albatross': [['bird', 'fly\_well']],

'animal\_type': [['cheetah']],

'animal\_type': [['zebra']],

'animal\_type': [['ostrich']],

'animal\_type': [['penguin']],

'animal\_type': [['albatross']],

}

facts = set()

def backward\_chaining(goal, kb, facts):

if goal in facts:

print(f"Goal '{goal}' is a known fact.")

return True

if goal in kb:

for rule in kb[goal]:

print(f"Trying to prove '{goal}' by proving rule conditions: {rule}")

rule\_proven = True

for condition in rule:

if not backward\_chaining(condition, kb, facts):

print(f"Condition '{condition}' failed for rule {rule}. Trying next rule.")

rule\_proven = False

break

if rule\_proven:

print(f"All conditions for rule {rule} proven. Goal '{goal}' is proven.")

if goal not in kb or not any(rule for rule in kb[goal]):

facts.add(goal)

return True

print(f"Cannot prove '{goal}'. Is it true? (yes/no):")

response = input().lower()

if response == 'yes':

facts.add(goal)

return True

else:

print(f"User stated '{goal}' is false.")

return False

print("Enter known facts (one per line, press Enter on an empty line to finish):")

while True:

fact = input().strip()

if not fact:

break

facts.add(fact)

print("\nEnter the goal you want to prove:")

target\_goal = input().strip()

print(f"\nAttempting to prove: {target\_goal}")

is\_proven = backward\_chaining(target\_goal, knowledge\_base, facts)

if is\_proven:

print(f"\nSuccessfully proven: {target\_goal}")

else:

print(f"\nCould not prove: {target\_goal}")

print("\nUpdated facts:", facts)

**OUT PUT:**

**Successfully proven: cheetah**

**Updated facts: {'has\_hair', 'has\_pointed\_teeth', 'has\_claws', 'has\_forward\_eyes', 'has\_spotted\_coat', 'mammal', 'eats\_meat', 'carnivore', 'cheetah'}**