# Implementation of Backward Chaining

## Aim:

To derive conclusions or solve problems by tracing back from the desired goal.

## Procedure:

1. **Define the Goal**: Identify the specific outcome or conclusion you want to achieve.
2. **Search Rules**: Look for inference rules that can lead to the goal.
3. **Check Conditions**: Verify if the conditions of the rules are satisfied by existing facts.
4. **Recursive Process**: If conditions are not met, treat them as sub-goals and repeat the process.
5. **Stop When Goal is Achieved**: The process ends when the goal is supported by facts or when all possibilities are exhausted.

## Python Implementation:

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

if goal in facts:

return True

for rule in rules:

if rule['conclusion'] == goal:

if all(backward\_chaining(condition, rules, facts) for condition in rule['conditions']):

facts.add(goal)

return True

return False

# Example rules and facts

rules = [

{'conditions': ['fever', 'cough'], 'conclusion': 'flu'},

{'conditions': ['headache', 'fever'], 'conclusion': 'migraine'}

]

facts = {'fever', 'cough'}

# Goal

goal = 'flu'

# Check if the goal can be achieved

result = backward\_chaining(goal, rules, facts)

print(f"Goal '{goal}' achieved: {result}”)

## Output:

Goal 'flu' achieved: True

# Result:

Thus the code successfully derived using the rules and facts provided.