# Experiment 5: Backward Chaining

## Aim

To implement backward chaining to prove a hypothesis.

## Procedure

1. Start with the goal.  
2. Search for rules that conclude the goal.  
3. Recursively prove all conditions of those rules.  
4. If all subgoals are proven, the goal is proven.

## Code

class BC:

def \_\_init\_\_(self, rules, facts):

self.rules = rules # {conclusion: [premises]}

self.facts = set(facts)

def prove(self, goal):

if goal in self.facts:

return True

for conc, prem in self.rules.items():

if conc == goal:

if all(self.prove(p) for p in prem):

self.facts.add(goal)

return True

return False

rules = {

'flies': ['has\_wings', 'is\_bird'],

'is\_bird': ['has\_feathers']

}

facts = ['has\_feathers']

bc = BC(rules, facts)

print(f"Can 'flies' be proven? {bc.prove('flies')}")

print(f"Known facts after proving: {bc.facts}")

## Output

