

## Implementation of Backward Chaining

### Aim

To implement a concise Backward Chaining algorithm to infer a goal using logical rules.

### Code

```
def backward_chaining(kb, goal, visited=None):
    visited = visited or set()
    if goal in kb['facts']: return True
    if goal in visited: return False
    visited.add(goal)
    return any(all(backward_chaining(kb, g, visited) for g in rule['if'])
               for rule in kb['rules'] if rule['then'] == goal)

kb = {
    'facts': ['Rain', 'Sprinkler'],
    'rules': [
        {'if': ['Rain'], 'then': 'WetGrass'},
        {'if': ['Sprinkler'], 'then': 'WetGrass'},
        {'if': ['WetGrass'], 'then': 'Slippery'}
    ]
}

goal = 'Slippery'
print(f"Can we conclude '{goal}'? =>", backward_chaining(kb, goal))
```

### Output

Can we conclude 'Slippery'? => True

### Result

The goal 'Slippery' is successfully inferred from the facts and rules using backward chaining.