ASSIGNMENT-2

PROBLEM - 1

Problem Statement

Sudoku is a logic-based puzzle played on a 9×9 grid divided into nine 3×3 subgrids. Each cell must contain a digit between 1–9 such that:

1. No digit repeats in any row.
2. No digit repeats in any column.
3. No digit repeats in any 3×3 subgrid.

We can view Sudoku as a Constraint Satisfaction Problem (CSP):

Variables: Each empty cell in the grid.
Domains: Possible digits (1–9).
Constraints: All-different constraints for rows, columns, and subgrids.

Initial Given Sudoku

The Sudoku puzzle provided below is a **partially filled 9×9 grid**.
Cells marked with **numbers (1–9)** are **pre-filled clues**, while cells marked with **"_"** represent **empty spaces** that students must fill according to Sudoku rules.

Interpretation:

- Each row, column, and 3×3 subgrid must contain **all digits from 1 to 9 exactly once**.
- The solver should respect the pre-filled digits and only assign values to blank cells that do not violate Sudoku constraints.

This puzzle serves as the **default input** for your solver.
Your implementation should also be able to handle other Sudoku puzzles (easy, medium, hard).

5	**3**	_	_	**7**	_	_	_	_
6	_	_	**1**	**9**	**5**	_	_	_
_	**9**	**8**	_	_	_	_	**6**	_
8	_	_	_	**6**	_	_	_	**3**
4	_	_	**8**	_	**3**	_	_	**1**
7	_	_	_	**2**	_	_	_	**6**
_	**6**	_	_	_	_	**2**	**8**	_
_	_	_	**4**	**1**	**9**	_	_	**5**
_	_	_	_	**8**	_	_	**7**	**9**

Implementation

1. Backtracking Search: Implement plain backtracking.
2. Forward Checking: Prune inconsistent values from domains after each assignment.
3. Heuristics:
    - MRV (Minimum Remaining Values)
    - Degree heuristic
    - LCV (Least Constraining Value)
4. Implement AC-3 (Arc Consistency).

PROBLEM-2

The Greek Philosopher Problem

You are given the following facts:
1. All men are mortal.
2. All Greeks are men.
3. All philosophers are thinkers
4. Socrates is a Greek and a philosopher.

You have to prove that :

Socrates is mortal
Using the Resolution Refutation method in First-Order Predicate Logic (FOL).

Instructions:
You must perform each transformation step of the reasoning process using Prolog. Create separate sections (or predicates) in Prolog that demonstrate every stage—from English sentences to automated resolution.
Show each resolution step and end with the empty clause (⊥) proving that the goal follows from
the given facts.

Expected Prolog File Structure:
Your .pl file should clearly demonstrate each phase:
% Step 1: Predicate logic representation
% Step 2: Negation of goal
% Step 3: Implication removal
% Step 4: Negation movement
% Step 5: Standardization
% Step 6: Skolemization
% Step 7: CNF clauses
% Step 8: Resolution implementation and execution

Deliverables:
1. A single Prolog file (greek_philosopher.pl) containing:
-The code and comments for all transformation steps.
-The resolution refutation process showing the derivation of the empty clause.
2. A screenshot of the final proof output (showing that the empty clause or contradiction has been derived).

PROBLEM-3

Delhi frequently experiences fluctuations in air quality due to heavy traffic, industrial activity, and varying meteorological conditions. These environmental factors contribute significantly to respiratory illnesses among residents. The task is to construct a Bayesian Belief Network that captures these dependencies and enables reasoning about how changes in causal factors (e.g., reduced traffic or lower industrial output) may influence downstream outcomes such as improved AQI or reduced hospital visits for respiratory diseases.

OBJECTIVE

Design and analyze a Bayesian Belief Network (BBN) that models the probabilistic relationships among environmental and health-related factors influencing air quality in Delhi. The model should represent causal dependencies between variables such as traffic, industrial emissions, weather, air quality index (AQI), and respiratory health outcomes. The exercise will emphasize conceptual modeling, data-driven parameter estimation, and multiple inference analyses.

TASKS

(a) Network Structure: Identify 5–6 relevant variables (for example, TrafficLevel, IndustrialActivity, Temperature, Humid- ity, AQI, and RespiratoryCases). Define appropriate causal links between them and represent the structure as a directed acyclic graph (DAG). Provide a short justification for each edge based on environmental reasoning.

(b) Joint Probability Representation: Express the joint probability distribution as a product of conditional probabilities following the BBN structure. Clearly specify conditional independence assumptions derived from the network.

(c) Parameter Estimation: Generate or collect a small dataset (minimum 200 samples) for the chosen variables. Estimate the conditional probability tables (CPTs) for each node using frequency-based methods. Use appropriate prior distributions, such as Dirichlet priors, for Bayesian parameter estimation to ensure smoother probability estimates.

(d) Inference and Queries: Perform probabilistic inference on your BBN to answer queries such as:

$$P\,(AQI = \text{good} \mid TrafficLevel = \text{low}) \quad \text{and} \quad P\,(RespiratoryCases = \text{low} \mid AQI = \text{good})$$

In addition, design and compute at least two more non-trivial inference queries of your choice (e.g., involving multiple evidence nodes or conditional probabilities such as $P\,(AQI = \text{poor} \mid IndustrialActivity = \text{high}, Temperature = \text{hot})$).

(e) Intervention Analysis: Evaluate the causal impact of external interventions using the do-operator. For example, compute:

$P\,(RespiratoryCases = \text{low} \mid do(TrafficLevel = \text{low}))$

and compare it with the corresponding observational probability. Discuss the implications in the context of environmental policy.

PROBLEM-4

Advanced Campus Cleaning Robot Planning (Obstacle-Free and Battery-Free Version)

Background

This assignment introduces students to multi-step automated planning using the Planning Domain Definition Language (PDDL). You will design and implement a cleaning robot capable of performing sequential cleaning tasks across multiple rooms on a campus.

Problem Statement

The Autonomous Cleaning Robot starts in the classroom. Both the classroom and cafeteria begin in a DIRTY state. Trash is also present in both areas. The dustbin area is used only for disposing of collected trash. The robot may choose ANY valid sequence to complete the cleaning task. It is NOT required to return to the dustbin after cleaning just one area; it may: • Clean both areas first and THEN dispose of trash, OR • Pick up and dispose of trash after each room. However, the final state MUST satisfy that all trash has been disposed of in the dustbin and the robot is not carrying trash at the end. The robot can carry only ONE unit of trash at a time. Therefore, if it collects trash from both rooms, it must visit the dustbin more than once

Environment Layout

Classroom ↔ Cafeteria ↔ Dustbin

Robot Capabilities

- Move between connected areas
- Sweep a dirty area (making it swept)
- Mop a swept area (making it fully cleaned)
- Pick up trash (only when not already carrying trash)
- Drop trash in the dustbin area

Action Preconditions and Effects

MOVE(from → to)

- Preconditions: The robot is at the 'from' location, and 'from' is connected to 'to'.
- Effects: The robot is now at 'to'. Nothing else about room cleanliness or trash changes.

SWEEP(location)

• Preconditions: The robot is at 'location' and the location is dirty (not yet swept or mopped).
• Effects: The location becomes swept and is no longer dirty. (It is ready for mopping.)

MOP(location)

• Preconditions: The robot is at 'location' and the location has already been swept.
• Effects: The location becomes mopped (fully cleaned).

PICK-TRASH(location)

• Preconditions: The robot is at 'location', there is trash at 'location', and the robot is not already carrying trash.
• Effects: The robot is now carrying one unit of trash, and the trash is removed from that location.

DROP-TRASH(dustbin)

• Preconditions: The robot is at the dustbin area and is currently carrying trash.
• Effects: The robot is no longer carrying trash; the trash is considered disposed of.

Goal State

• Both the classroom and cafeteria are fully mopped (cleaned).
• All trash from both areas is collected and disposed of in the dustbin.
• The robot is not carrying any trash.

Formal Goal Representation: (and (mopped classroom) (mopped cafeteria) (not (carrying-trash robot1)))

Starting and Ending Location Clarification

• The robot STARTS in the classroom.
• The robot DOES NOT need to end in a specific location; only the goal conditions must be satisfied.

Planner Execution

Use any PDDL planner (Fast Downward / Planning.Domains online editor / Metric-FF) to run your model and generate the final action plan. Record the generated plan steps in a separate text file.

Report

Create a short explanation (3–5 sentences) describing how the generated plan achieves the final goal.

Deliverables

- cleaning_domain.pddl : Domain definition.
- cleaning_problem.pddl : Initial and goal state description.
- output_plan.txt : Generated plan.
- CleaningRobot_Planning_Report.pdf : One-page summary.