Assignment No: - 2

Name :- Isha Ghorpade

Enrollnment :- 22420020

Roll :- 381066

Problem Statement

Implement a Constraint Satisfaction Problem (CSP) where variables, domains, and constraints are defined, and a solution is found using backtracking search.

Objectives

- To understand the formulation of Constraint Satisfaction Problems.
- To represent problems in terms of variables, domains, and constraints.
- To implement CSP using backtracking.
- To analyze real-life applications of CSPs in AI.

Theory

What is a CSP?

A Constraint Satisfaction Problem (CSP) is a mathematical problem defined by:

1. Variables (X): A set of unknowns to be assigned values.
   Example: Regions on a map (A, B, C, D).
2. Domains (D): A set of possible values for each variable.
   Example: Colors {Red, Green, Blue}.
3. Constraints (C): Restrictions that limit possible assignments.
   Example: Adjacent regions must not have the same color.

Formally, a CSP is defined as a triple:

CSP=(X,D,C)CSP = (X, D, C)

CSP Representation

- State Space: All possible assignments of values to variables.
- Solution: An assignment of values to all variables that satisfies all constraints.
- Example: Map coloring where no two adjacent states share the same color.

CSP Solving Strategies

1. Backtracking Search
    - Assign values one by one.
    - Backtrack if constraints are violated.
    - Simple but can be slow for large CSPs.
2. Forward Checking
    - After assigning a value, eliminate inconsistent values from other variables.
    - Reduces unnecessary search.
3. Arc Consistency (AC-3 Algorithm)
    - Ensures that for every variable and value, there exists a consistent assignment in neighboring variables.

Example: Map Coloring Problem

- Variables: {A, B, C, D}
- Domains: {Red, Green, Blue}
- Constraints: Adjacent nodes must not share the same color.

Methodology

1. Define variables, domains, and constraints.
2. Start with an empty assignment.
3. Select a variable and assign a value.
4. Check if constraints are satisfied.
5. If consistent, move to the next variable.
6. If not, backtrack and try another value.
7. Repeat until all variables are assigned valid values.

Advantages

- Provides a systematic approach to solving problems.
- Easily applicable to many real-life problems.
- Can be optimized using heuristics and consistency checks.

Limitations

- Backtracking can be slow for large problems.
- May require advanced techniques (like heuristics) for efficiency.
- Complex CSPs may still be computationally expensive.

Applications

- Map coloring and graph coloring problems.
- Scheduling tasks (exams, employees, flights).
- Resource allocation in projects.
- Sudoku and crossword puzzles.
- AI planning and configuration problems.

Algorithm (Backtracking for CSP)

1. If all variables are assigned → return solution.
2. Select an unassigned variable.
3. For each value in the domain:
    - If consistent with constraints → assign and continue.
    - Else backtrack.
4. Repeat until solution is found or no assignment is possible.

Conclusion

Constraint Satisfaction Problems (CSPs) provide a flexible framework to solve real-world problems where multiple variables interact under restrictions. By using backtracking and constraint checking, CSPs can be solved efficiently for small and medium-sized problems. For larger and more complex problems, techniques like forward checking, arc consistency, and heuristics make CSP solving practical and scalable.