Assignment 1:- State space representation

Group name : L

Prepared by:

Praval Pattam	B220057CS	praval_b220057cs@nitc.ac.in
Theenesh Potluri	B221121CS	potluri_b221121cs@nitc.ac.in
Pranav Sai Sarvepalli	B220055CS	pranav_b220055cs@nitc.ac.in
Venkatesh Banoth	B220585CS	venkatesh_b220585cs@nitc.ac.in

Instructor:	Dr Lijiya A Dr Jay Prakash Dr Shweta Dr Chandramani Chaudhary
Course:	CS4023D Artificial Intelligence
Due date:	22/08/2025

Contents

1 Problem Statement: Class Scheduling as a CSP

The problem is to assign a set of university classes to time slots and rooms without violating any hard constraints. This is a classic Constraint Satisfaction Problem (CSP) as defined in ***Artificial Intelligence: A Modern Approach (AIMA)***.

Given:

- A set of **Classes**: C={ C1 ,C2 ,..., Cn }
- A set of **Time Slots**: T={ T1, T2 ,..., Tm}
- A set of **Rooms**: R={ R1, R2,..., Rk}
- For each class Ci, a **Professor** Pi and a list of enrolled **Students** Si.
- For each room Rj, a **Capacity** Capj.

Constraints:

The assignment must satisfy the following absolute constraints:

1. **No Room Double-Booking:** Two different classes cannot be assigned to the same room at the same time.
   - *Form:* Alldiff (Room (Ci, Tj)) for a fixed Tj.
2. **No Professor Conflict:** A professor cannot teach two classes simultaneously.
   - *Form:* For any two classes Ca and Cb where Pa=Pb, T(Ca)≠T(Cb).
3. **No Student Conflict:** A student cannot attend two classes simultaneously.
   - *Form:* For any two classes Ca and Cb with overlapping students
   - (Sa ∩ Sb ≠ ∅), T(Ca) ≠ T(Cb).
4. **Room Capacity:** The number of students in a class must not exceed the capacity of the room it is assigned to.
   - *Form:* For every assignment, |Si| ≤ Cap(R(Ci)).

Objective:

Find a complete assignment of all classes to a time slot and a room such that all constraints are satisfied.

2 State Space Representation

Following the standard CSP formulation in AIMA , the problem is defined by the triple (X,D,C):

- **Variables (X):** The set of variables is the set of classes to be scheduled.
  X={ C1, C2, C3,..., Cn } Each variable Ci represents one class.
- **Domains (D):** The domain of each variable is the set of all possible resources (time slot and room pairs) it can be assigned to.
  For each class Ci, Di={(Tj, Rk) | Tj ∈ T,Rk ∈ R}
  Initially, every resource pair is a possible value for every class.
- **Constraints (C):** The constraints are the rules listed in the problem statement. They specify the allowable combinations of assignments. Formally, they can be expressed as:
  1. **Constraint 1 (Room/Time):** For a given time slot Tj and room Rk, only one class can be assigned the value (Tj,Rk).
  2. **Constraint 2 (Professor):** For a given professor P and time slot Tj, only one class taught by P can be assigned a value containing Tj (i.e., (Tj, Rk) for any room Rk).
  3. **Constraint 3 (Student):** For a given student S and time slot Tj only one class containing S can be assigned a value containing Tj.
  4. **Constraint 4 (Capacity):** For a class Ci and a room Rk, the assignment (Tj, Rk) is only allowed if |Si| ≤ Capk . This is a unary constraint that can be enforced by pre-processing the domains, removing any (Tj,Rk) where the room capacity is insufficient for the class.

Viewing the CSP as a Search Problem (for BFS, DFS, etc.):

- **States:** A state is a partial assignment of values (time/room pairs) to variables (classes). The assignment must not violate any constraints.
- **Initial State:** The empty assignment, where no variables have been assigned a value.
- **Actions:** Assign a value from its current domain to an unassigned variable, provided the assignment does not violate any constraints.
- **Transition Model:** Returns a new state where the chosen variable has the chosen value, and the domains of future variables may be reduced due to forward checking (if implemented).
- **Goal Test:** The assignment is complete (all variables are assigned a value). Since actions are only taken if they are consistent, a complete assignment is a solution.
- **Path Cost:** Not applicable for a pure satisfaction problem. The cost of a path is typically the number of assignments (1 per step), and the goal is to find any path to a goal state. For optimization, a cost function (e.g., prioritizing certain time slots) would be defined.