FUNDAMENTALS OF ARTIFICIAL INTELLIGENCE
Lab Exercise – 2

Name : ADVAIT GURUNATH CHAVAN
Email ID : advaitchavan135@gmail.com

CDAC – NOIDA
PGDAI

ASSIGNMENT NO 2

Name : ADVAIT GURUNATH CHAVAN
Email ID : advaitchavan135@gmail.com

Q1. Chess Agents :

Consider two intelligent agents playing chess with a clock. One of them is called "Deep Blue" while the other is called Gary Kasparov.

Specify the task environment in terms of "PEAS"

Specify the Properties of this task environment.

1.Performance Measure:
- Winning the game (checkmate or opponent's resignation).
- Maximizing material advantage and positional strength.
- Playing within the time limit.
- Minimizing errors and blunders.

2.Environment:
- The chessboard (8x8 grid).
- Pieces with predefined legal moves.
- Time constraints (chess clock).
- Opponent's moves.

3.Actuators:
- Moving pieces on the board.
- Controlling the clock (making moves within the time).
- Capturing opponent's pieces.

4.Sensors:
- Board state (positions of pieces).
- Move history (previous moves made).
- Timer readings.

Properties of the Task Environment:

1.Fully Observable vs. Partially Observable:
- **Fully Observable**: The entire chessboard and game state are visible at all times.

- **Deterministic vs. Stochastic:**
  - **Deterministic**: The outcome of a move is completely predictable.

1.Episodic vs. Sequential:
- **Sequential**: Each move affects future decisions and the final outcome.

2.Static vs. Dynamic:
- **Semi-Dynamic**: The board does not change unless a player makes a move, but the clock adds a time-dependent factor.

3.Discrete vs. Continuous:
- **Discrete**: A finite number of possible moves and positions.

4.Single-agent vs. Multi-agent:
- **Multi-agent**: Two intelligent agents (Deep Blue and Garry Kasparov) compete against each other.

Q2. Design State space for agent which:

- Can move in four directions up, down, right and left
- Can sense heat in cell on left or right or up or down
- Environment is rectangle of size n by m
- Environment has agent, a lion and home
- Goal of the agent is to reach home
- All cells adjacent to lion have heat
- If Agent goes to adjacent cell to lion then agent loses

Write conditions for movement precisely and design the space tree

Name : ADVAIT GURUNATH CHAVAN
Email ID : advaitchavan135@gmail.com

Agent(A)	(1,0)	(2,0)	(3,0)
(0,1)	(1,1)	(2,1)	(3,1)
(0,2)	(1,2)	Lion(L)	(3,2)
(0,3)	(1,3)	(2,3)	Home(H)

The **state space** consists of:
•The **position of the agent**: (Ax,Ay)
•The **position of the lion**: (Lx,Ly)
•The **position of the home**: (Hx,Hy).

	Heat Zones Adjacent to Lion that must be avoided by the agent else the agent loses the game.
	Permissible area where the agent can move freely, without any danger

Movement Conditions:
The agent can move **up, down, left, or right** under these constraints:

1.Boundary Conditions:

- The agent cannot move outside the grid.
- Allowed moves:
  - **Up**: (x,y)→(x,y+1) if y+1<m ; m → Number of rows
  - **Down**: (x,y)→(x,y−1) if y−1≥0
  - **Left**: (x,y)→(x−1,y) if x−1≥0
  - **Right**: (x,y)→(x+1,y if x+1<n; n → Number of Columns

2.Heat Sensing:

- If the agent is at (x,y) and any of the adjacent cells contain heat (meaning a lion is nearby), the agent should **avoid** moving into that cell.

3.Game Losing Condition:

- If the agent moves to an **adjacent cell of the lion**, the game is lost.

4.Goal State:

- The agent wins when it reaches the **home position (Hx,Hy).**

Write conditions for movement precisely and design the space tree

Q3. The missionaries and cannibals problem is usually stated as follows. Three missionaries and three cannibals are on one side of a river, along with a boat that can hold one or two people. Find a way to get everyone to the other side without ever leaving a group of missionaries in one place outnumbered by the cannibals in that place. This problem is famous in AI because it was the subject of the first paper that approached problem formulation from an analytical viewpoint (Amarel, 1968). Formulate the problem precisely, making only those distinctions necessary to ensure a valid solution. Draw a diagram of the complete state space.

Name : ADVAIT GURUNATH CHAVAN
Email ID : advaitchavan135@gmail.com

State Representation

Each state can be represented as **(M, C, B)** where:

- **M** = Number of missionaries on the left side.
- **C** = Number of cannibals on the left side.
- **B** = Position of the boat (0 = left, 1 = right).

The goal state is **(0, 0, 1)** (all missionaries and cannibals on the right side).

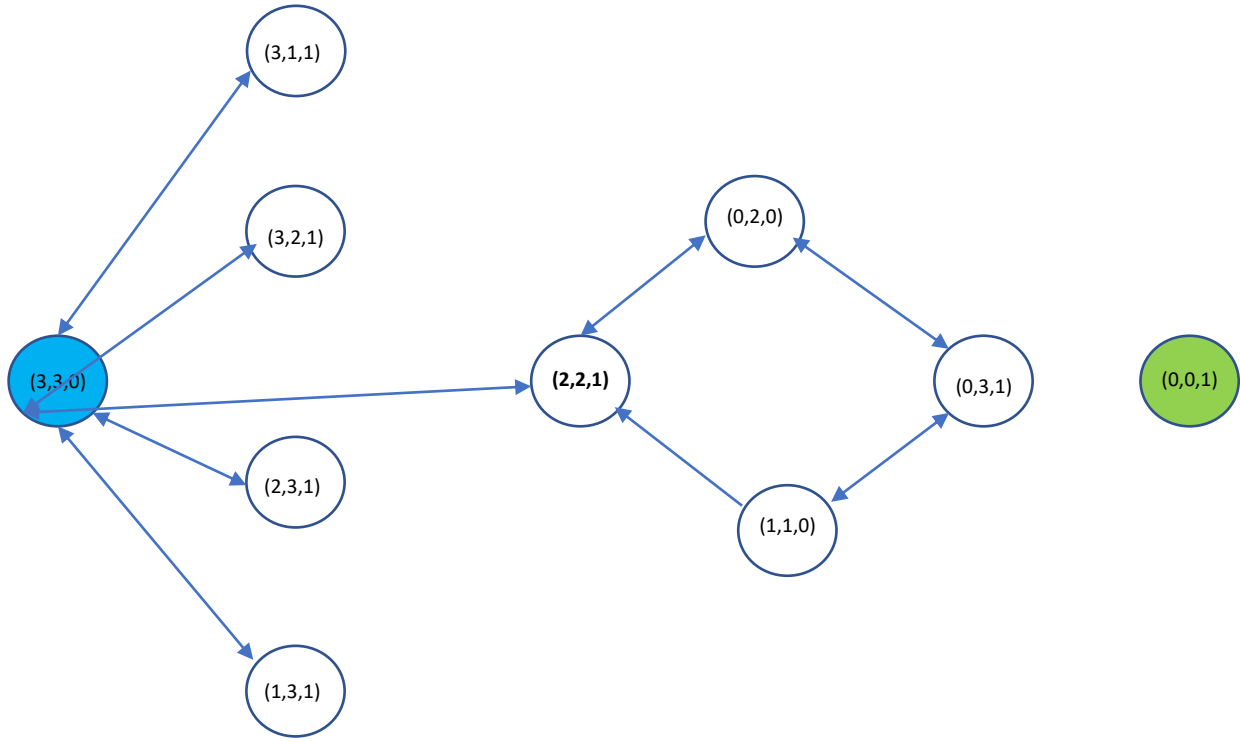
Valid Moves

The boat can carry:

1. One missionary (**1M**)
2. One cannibal (**1C**)
3. Two missionaries (**2M**)
4. Two cannibals (**2C**)
5. One missionary and one cannibal (**1M1C**)

Constraints (Safe Conditions)

1. **Missionaries should never be outnumbered by cannibals on either side**, i.e.,
    1. If M>0 then M≥C on both sides.
2. The boat **cannot cross** if it's empty.

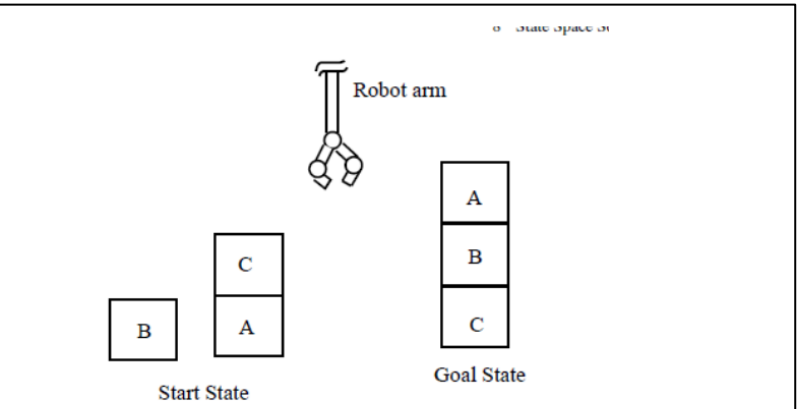


Blue Node: Start state (3,3,0)

Green Node: Start state (0,0,1)

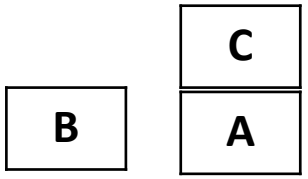
Q4. Draw a diagram of the complete state space for block world problem in AI.

Name : ADVAIT GURUNATH CHAVAN
Email ID : advaitchavan135@gmail.com

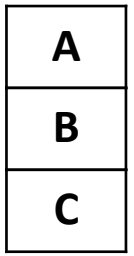


Assume that following rules for moves will be followed by the robot arm for carrying out this job:

- stack(x, y): stack block x on block y,
- lift(x): lift-up the block x,
- putg(x): put block x on ground,
- unstack(x, y): unstack block x from block y.



Start State



Goal State

Each state represents the **positions of the blocks**. The robot arm moves one block at a time.

1.Start State

- B on ground, A on ground, C on A.

2.Possible Moves from Start State:

- **Unstack(C, A)** → C is now in hand.
- **Lift(B)** → B is now in hand.
- **Lift(A)** → A is now in hand.

3.After Unstack(C, A):

- C is in hand, A on ground, B on ground.

4.After Placing C on Ground :

- C on ground, A on ground, B on ground.

5.Lift A and Place on B:

- A is now on B.

6.Lift C and Place on Ground:

- C is already on the ground, so no change.

7.Lift B and Place on C:

- Now we have A on B, and B on C.

8.Goal State Achieved

- **A is on B, B is on C, and C is on the ground.**