3rd December, 2024. Tuesday. Laboratory - 9

Q: Create a knowledge base consisting of first order logic statements and prove the given query using forward reasoning.

Forward chaining is one of the two methodologies using an inference engine which starts with a base state and uses the inference rules and available knowledge in the forward direction till it reaches the goal state.

As per law, it is crime for an American to sell weapons to hostile nation. Country A, enemy of America, has some missiles, and all the missiles were sold it by Robert, who is an American citizen."

Prove that "Robert is criminal."

Solution: / Proof /

→ It is a crime for an American to sell weapons to hostile nation
Lets say p, q, and r are variables.

American(p) ∧ weapon(q) ∧ sells (p, q, r) ∧
Hostile(r) ⇒ Criminal(p)

→ Country A has some missiles

∃ x owns (A, x) ∧ missile (x)
Existential Instantiation - thus introducing a new constant T
Owns (A, T)
missile (T)

→ All of the missiles were sold to country A by Robert
∀x Missile(x) ∧ Owns(A, x) ⇒ sells (Robert, x, A)

→ Missiles are weapons

$Missile(x) \Rightarrow Weapon(x)$

→ Enemy of America is known as hostile

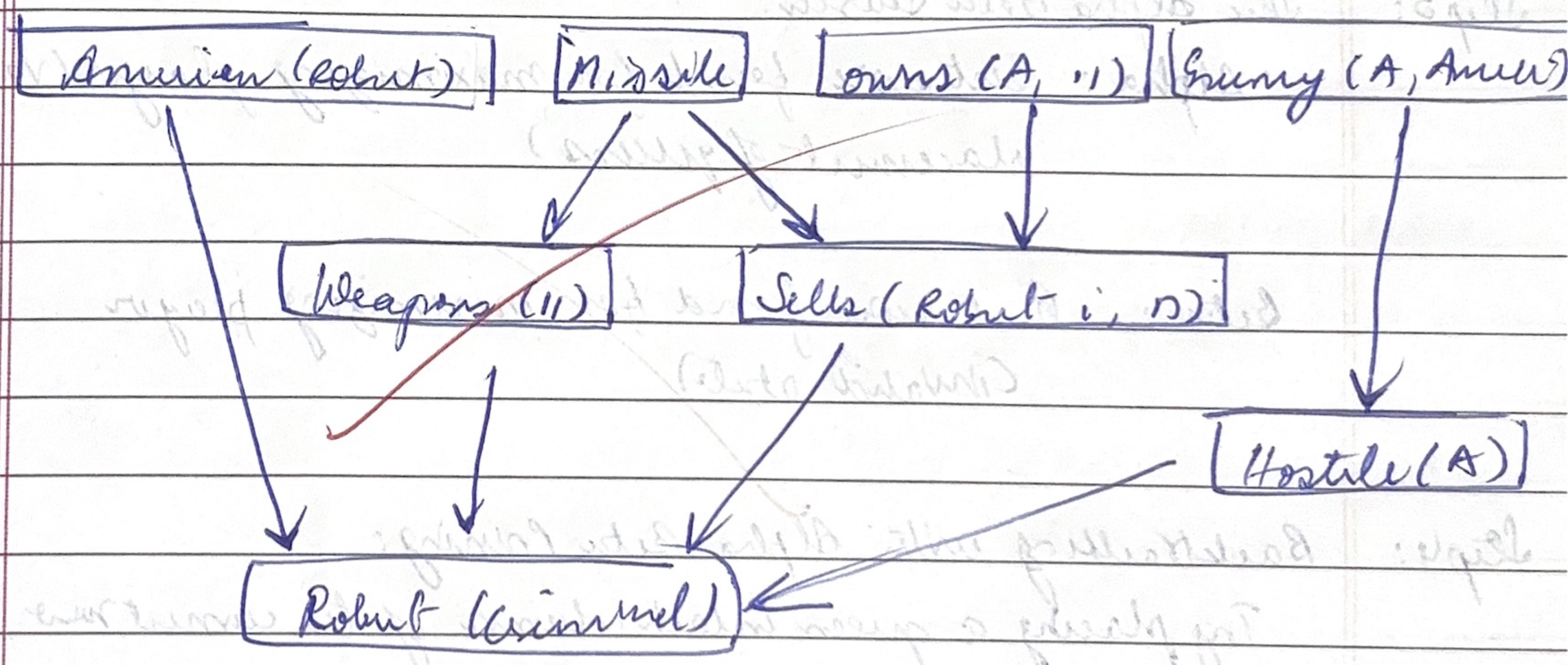
$\forall x\ Enemy(x, America) \Rightarrow Hostile(x)$

→ Robert is an American

$American(Robert)$

→ The country A, an enemy of America

$Enemy(A, America)$



Date
Page

Q: 8 Queens Problem: Alpha Beta Pruning.

Algorithm:

Step 1: Initialize the board: Create an 8x8 board where each row will hold the column number of the queens placed in that row. Start with no queens placed: (board = [-1] * 8)

Step 2: Define 'is_safe' function: check if the placing of a queen at (row, column) is safe; i.e the two queens threaten each other.

Step 3: The Alpha Beta Search:
- Alpha: Best score found for maximizing player (valid placement of queens)

Beta: Best score found for minimizing player (invalid state)

Step 4: Backtracking with Alpha Beta Pruning:
Try placing a queen in each column of the current row.
Recursively place queens in subsequent rows.
Prune branches where further exploration is unnecessary (based on alpha and beta values)

Step 5: Return Solution: If solution is found, return the solution

Date
Page

Q: Minmax Algorithm for Tic Tac Toe

Step 1: Check if the game is over:

- If a player wins, return the score
  - +1 for 'X' win
  - -1 for 'O' win
  - 0 for a draw

If the board is full and no one has won, return (0' draw)

Step 2: Maximizing player (X)

- Initialize the best score to $-\infty$
- For each available move on the board
  - → Make the move (place 'X')
  - → Call the minmax function recursively, switching to the minimizing player (O)
- Undo the move
- Update the best score with the maximum value from the recursive call.
- Return the best score

Step 3: Minimizing Player (O)

- Initialize the best score to ∞
- For each available move on the board.
  - Make the move (place 'O')
  - Call the minmax function recursively.
- Similar to Step 2

Sneha SB
3/12/24

Step 4: Find the best move for the maximizing player (X).

- For each available move, evaluate move using minmax
- Return the move with highest score