

so, prove Robert is a criminal

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

$p, q, x$  are variables

America can't sell weapons to hostile nations.  
America  $(p) \wedge$  weapons  $(q) \wedge$  sell  $(p, q, x) \wedge$  Hostile  $(x)$   
 $\Rightarrow$  Criminal  $(p)$

Country A has missiles

$\exists x$  Owns  $(A, x) \wedge$  Missiles  $(x)$   
 $x$  is an object (here missile)

consider a constant  $(T_1)$

owns  $(T_1)$

Missile  $(T_1)$

$\forall x$  Missile  $(x) \wedge$  Owns  $(A, x) \Rightarrow$  Sells  $(Robert, x, A)$

All missiles are sold to A by Robert  
which means the missile  $T_1$  was also sold to A by Robert.

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

If  $x$  is a missile, it implies  $x$  is also a weapon

$\forall x \text{ Enemy}(x, \text{America}) \Rightarrow \text{Hostile}(x)$

If  $x$  is an enemy to America, it implies  $x$  is hostile to America

if  $p$  is Robert then America (Robert)  
Robert is an American

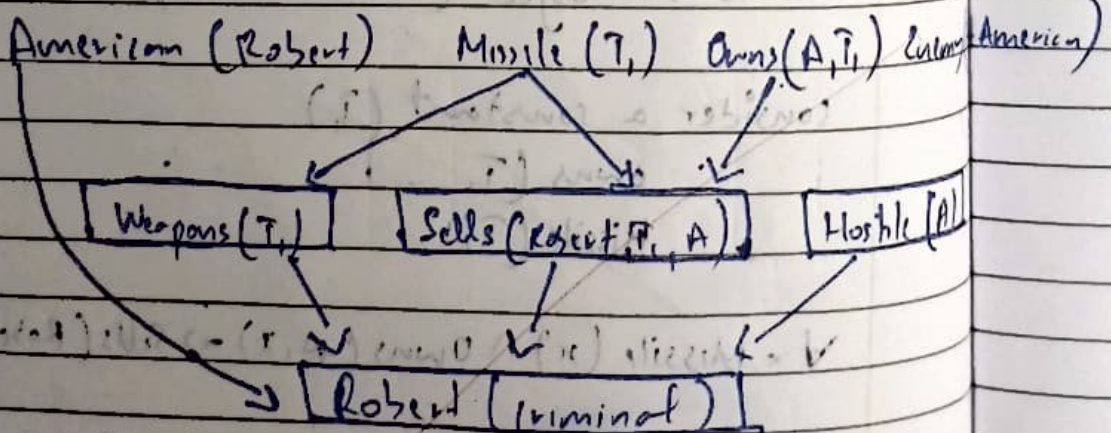
and  $A$  is an enemy of America

$\text{Enemy}(A, \text{America}) \Rightarrow \text{Hostile}(A)$

$\therefore \text{America}(\text{Robert}) \wedge \text{Weapon}(T_1) \wedge \text{Sells}(\text{Robert}, T_1, A) \wedge \text{Hostile}(A)$

$\therefore \text{Criminal}(\text{Robert})$

forward chaining





# Tic Tac Toe Using Min Max

Define 3x3 grid where each cell can be  
 X: player 1  
 O: player 2

+10 : Maximize wins for board  
 -10 : Minimize wins  
 0 : Draw

def minimax(board, depth, is\_maximize):  
 score = evaluate(board)

if score == 10 or score == -10  
 return score  
 if is board full (board)  
 return 0

if is Maximize:

best = -infinity

for each empty cell:

board[cell] = 'X'

best = max(best, minimax(board, depth+1, False))

board[cell] = empty  
 return best

else:

if is Minimize:

best = +infinity  
 for each empty cell

makemove (board, cell, 'x')

$$\text{Score} = \min_{\text{max}} (\text{board}, \text{depth}+1, \text{value})$$

undo-mot. (bocet; cell')

$$\text{best slow} = \min(\text{best slow}, \text{slow})$$

return bestScore: 6.00 2.00 2.00

```
def findBestmove (board):
```

best move = None

best value  $\propto$  infinity

for each empty cell in board

make more (6012, cell, 'x')

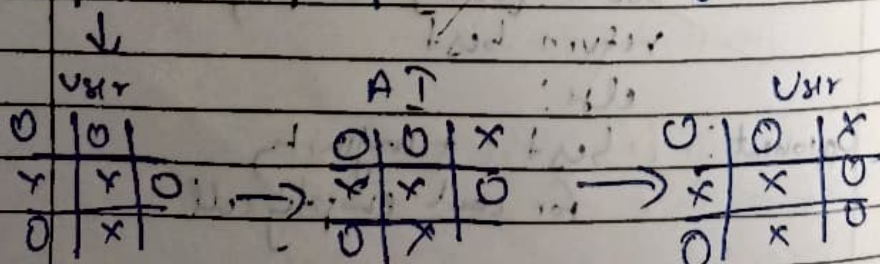
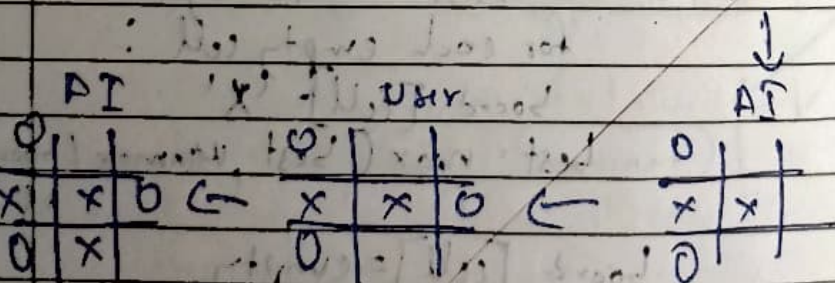
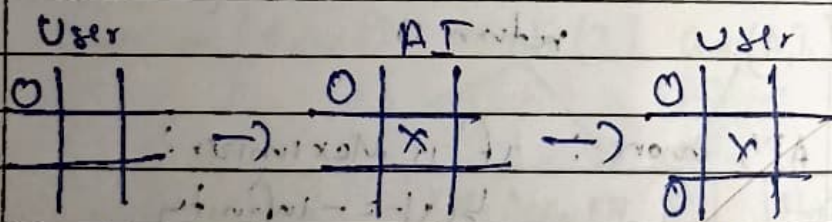
more reliable - insurance (bond, id, etc.)

if move value > best value

best move = cell

best value = move value

return best move



It is a draw!