

A **value** is -1, 0, or 1, corresponding to a loss, tie, or win for player 'o'.

```
def [val: gameState -> value]
```

```
val(S) =  
    1 if xwins(S);  
    -1 if owins(S);  
    0 if tiegame(S);  
    max({val(x) | x in successors(S)} if osMove(S);  
    min({val(x) | x in successors(S)} if xsMove(S);
```

You will need to write the following helper functions. They may call functions you have written previously, such as *threeInRow*, *newXsOs*, *boardFull*, etc.

- *xwins(S)* iff x has won the game in state S.
- *owins(S)* iff o has won the game in state S.
- *tieGame(S)* iff the game has ended in a tie in state S.
- *osMove(S)* iff the game is not over in state S and it is o's move.
- *xsMove(S)* iff the game is not over in state S and it is x's move.
- *successors(S)* is the set of all game states that may be obtained from S by making a single move which is legal in S.

The definition of *successors* uses list comprehension. You cannot use set comprehension for it. You can copy the code below for *successors*:

```
# successors: gameState -> set<gameState>  
#  
# successors(S) is the set of all game states that may be obtained  
# from S by making a single move which is legal in S.  
  
def successors(S):  
    xs = S[0]  
    os = S[1]  
    return [newXsOs(xs,os,c) for c in range(1,10) if legalMove(c,xs,os)]
```

Finally, using *value*, we can define the function that chooses the hard computer move. You can use my code for that.:

```
# hComputerMove : position*position -> cell
# hComputerMove(xs,os) is a move with the highest value the computer
# can force in the end state from game state (xs,os).

def hComputerMove(xs,os) =
  Mvs = [c for c in range(1,10) if legalMove(c,xs,os)]
  wins  = [c for c in Mvs if val(newXsOs(xs,os,c)) == 1]
  ties  = [c for c in Mvs if val(newXsOs(xs,os,c)) == 0]
  losses = [c for c in Mvs if val(newXsOs(xs,os,c)) == -1]
  return (wins+ties+losses)[0]
```