

Inteligência Artificial 04/05 — Aula 6

1. Crie um predicado `getadj(+X,+Y,-ListAdj)` que obtém uma lista `ListAdj` com as posições `(Col,Lin)` adjacentes a `(X,Y)`, com base num tabuleiro de 7x7 cujo canto superior esquerdo corresponde a `(1,1)`. Exemplos:

```
?- getadj(3,2,ListAdj).  
ListAdj = [(2,1),(3,1),(4,1),(2,2),(4,2),(2,3),(3,3),(4,3)] ? ;  
no
```

```
?- getadj(2,7,ListAdj).  
ListAdj = [(1,6),(2,6),(3,6),(1,7),(3,7)] ? ;  
no
```

```
?- getadj(0,2,ListAdj).  
no
```

2. No mesmo molde do exercício anterior, crie um predicado `getadj2(+X,+Y,-ListAdj)` que obtém uma lista `ListAdj` com as posições `(Col,Lin)` que se encontram a uma distância de dois de `(X,Y)`. Exemplos:

```
?- getadj2(3,2,ListAdj).  
ListAdj = [(1,1),(5,1),(1,2),(5,2),(1,3),(5,3),(1,4),(2,4),(3,4),  
           (4,4),(5,4)] ? ;  
no
```

```
?- getadj2(2,7,ListAdj).  
ListAdj = [(1,5),(2,5),(3,5),(4,5),(4,6),(4,7)] ? ;  
no
```

```
?- getadj2(0,2,ListAdj).  
no
```

3. Crie um predicado `canMove(+X1,+Y1,+X2,+Y2,+Board)` que sucede se o jogador 'x' puder jogar de (X1,Y1) para (X2,Y2), de acordo com a informação contida no tabuleiro Board. Exemplos:

```
?- canMove(1,1,3,3,['x','.','.','.','.','.','.','.','o'],
                  ['.','.','.','.','.','.','.','.'],
                  ['.','.','.','.','.','.','.','.'],
                  ['.','.','.','.','.','.','.','.'],
                  ['.','.','.','.','.','.','.','.'],
                  ['.','.','.','.','.','.','.','.'],
                  ['o','.','.','.','.','.','.','x']]).
```

yes

```
?- canMove(1,1,3,3,['o','.','.','.','.','.','.','x'],
                  ['.','.','.','.','.','.','.','.'],
                  ['.','.','.','.','.','.','.','.'],
                  ['.','.','.','.','.','.','.','.'],
                  ['.','.','.','.','.','.','.','.'],
                  ['.','.','.','.','.','.','.','.'],
                  ['x','.','.','.','.','.','.','o']]).
```

no

```
?- canMove(1,1,1,1,['x','.','.','.','.','.','.','o'],
                  ['.','.','.','.','.','.','.','.'],
                  ['.','.','.','.','.','.','.','.'],
                  ['.','.','.','.','.','.','.','.'],
                  ['.','.','.','.','.','.','.','.'],
                  ['.','.','.','.','.','.','.','.'],
                  ['o','.','.','.','.','.','.','x']]).
```

no

4. Crie um predicado `allMoves(+X,+Y,+Board,-ListMoves)` que obtém uma lista `ListMoves` com todos os movimentos possíveis a partir de (X,Y) para o jogador 'x', de acordo com a informação contida no tabuleiro Board. Exemplos:

```
?- allMoves(1,1,['x','.','x','.','.','.','.','o'],
             ['o','o','o','.','.','.','.'],
             ['o','.','o','.','.','.','.'],
             ['.','.','.','.','.','.','.'],
             ['.','.','.','.','.','.','.'],
             ['.','.','.','.','.','.','.'],
             ['o','.','.','.','.','.','.','x']],ListMoves).
```

```
ListMoves = [(2,1),(2,3)] ? ;
```

no

```
?- allMoves(2,2,[[ 'x','.', 'x','.','.','.','.', 'o'],
                  [ 'o', 'o', 'o','.','.','.','.', '.'],
                  [ 'o','.', 'o','.','.','.','.', '.'],
                  [ '.','.','.','.','.','.','.', '.'],
                  [ '.','.','.','.','.','.','.', '.'],
                  [ '.','.','.','.','.','.','.', '.'],
                  [ 'o','.','.','.','.','.','.', 'x']],ListMoves).

no
```

```
?- allMoves(0,2,[[ 'x','.', 'x','.','.','.','.', 'o'],
                  [ 'o', 'o', 'o','.','.','.','.', '.'],
                  [ 'o','.', 'o','.','.','.','.', '.'],
                  [ '.','.','.','.','.','.','.', '.'],
                  [ '.','.','.','.','.','.','.', '.'],
                  [ '.','.','.','.','.','.','.', '.'],
                  [ 'o','.','.','.','.','.','.', 'x']],ListMoves).

no
```

5. Crie um predicado `setxy(+X,+Y,+Board,+Cell,?NewBoard)` especializado para tabuleiros 7x7 que sucede sse (X,Y) é uma posição válida do tabuleiro Board e NewBoard é o tabuleiro que resulta de Board actualizando a posição (X,Y) com o conteúdo Cell. Exemplos:

```
?- setxy(2,1,[[ 'x','.', 'x','.','.','.','.', 'o'],
               [ 'o', 'o', 'o','.','.','.','.', '.'],
               [ 'o','.', 'o','.','.','.','.', '.'],
               [ '.','.','.','.','.','.','.', '.'],
               [ '.','.','.','.','.','.','.', '.'],
               [ '.','.','.','.','.','.','.', '.'],
               [ 'o','.','.','.','.','.','.', 'x']], 'x',NewBoard).

NewBoard = [[ 'x', 'x', 'x','.','.','.','.', 'o'],
            [ 'o', 'o', 'o','.','.','.','.', '.'],
            [ 'o','.', 'o','.','.','.','.', '.'],
            [ '.','.','.','.','.','.','.', '.'],
            [ '.','.','.','.','.','.','.', '.'],
            [ '.','.','.','.','.','.','.', '.'],
            [ 'o','.','.','.','.','.','.', 'x']] ? ;

no
```

6. Crie um predicado `makeMove(+X1,+Y1,+X2,+Y2,+Board,?NewBoard)` que sucede se o tabuleiro `NewBoard` for o resultado do jogador 'x' jogar de `(X1,Y1)` para `(X2,Y2)` a partir do tabuleiro `Board`. Exemplos:

```
?- makeMove(1,1,2,3,['x','.','x','.','.','.','.','o'],
                  ['o','o','o','o','.','.','.'],
                  ['o','.','o','.','.','.'],
                  ['.','.','.','.','.','.'],
                  ['.','.','.','.','.'],
                  ['.','.','.','.'],
                  ['o','.','.','.','.','x']),NewBoard).
NewBoard = [['.','.', 'x','.','.','.','o'],
            ['x','x','x','o','.','.'],
            ['x','x','x','.','.'],
            ['.','.','.','.'],
            ['.','.','.'],
            ['.','.'],
            ['o','.','.','.','x']] ? ;
no
```

```
?- makeMove(2,2,2,1,['x','.','x','.','.','.','o'],
                  ['o','o','o','o','.','.'],
                  ['o','.','o','.','.'],
                  ['.','.','.'],
                  ['.','.'],
                  ['.'],
                  ['o','.','.','.','x']),NewBoard).
no
```

```
?- makeMove(2,3,1,1,['x','.','x','.','.','.','o'],
                  ['o','o','o','o','.','.'],
                  ['o','.','o','.'],
                  ['.','.'],
                  ['.'],
                  ['.'],
                  ['o','.','.','.','x']),NewBoard).
no
```