



Escola de Ciências e Tecnologia

Departamento de Informática

Licenciatura de Engenharia Informática

Unidade curricular de Inteligência Artificial

Ano letivo 2020/2021

Relatório

3º Trabalho de Inteligência Artificial

Docentes

Professora Irene Pimenta Rodrigues

Discentes

José Santos nº43017

Ludgero Teixeira nº41348

Pedro Claudino nº39870

Évora, Maio de 2021

Exercício 1

- a) Escolha uma estrutura de dados para representar os estados do jogo

```
estado_inicial(e(1,3,4,4)).
```

- b) Defina o predicado `terminal(estado)` que sucede quando o estado é terminal.

```
terminal(e(0,0,0,0)).
```

- c) Defina uma função de utilidade que para um estado terminal que deve ~ retornar o valor do estado (ex: -1 perde, 0 empata, 1 ganha)

```
valor(E,1,P):- terminal(E), X is P mod 2, X=0,!.  
valor(E,-1,_):- terminal(E).
```

- d) Use a implementação da pesquisa **minimax** dada na aula prática para escolher a melhor jogada num estado.

```
tira(4,2)
```

```
1 - |  
2 - | | |  
3 - | | | |  
4 - | |
```

```
estado_inicial(e(1,3,4,4)).
```

```
g(Jogo):- [Jogo], estado_inicial(Ei), minimax_decidir(Ei,0p, Ess),  
write(0p),nl,(Jogo=nim->escNim(Ess);escWQueen(Ess)),nl.
```

- e) Implemente a pesquisa **Alfa-Beta** e compare os resultados (tempo e espaço).

Não conseguimos implementar a pesquisa Alfa-Beta.

- f) Defina uma função de avaliação que estime o valor de cada estado do jogo use os dois algoritmos anteriores com corte em profundidade e compare os resultados (tempo e espaço).

Não conseguimos implementar a uma função de avaliação.

- g) Implemente um agente inteligente que joga o jogo do Nim.

```
jogar(Jogo):- [Jogo], estado_inicial(E1),
    write('Inicio - '),nl,
    (Jogo=nim -> escNim(E1),agenteNim(E1); escWQueen(E1),agenteWQueen(E1)).

it(Ei, Ess):- minimax_decidir(Ei, Opf, Ess).

ganhou(-1):- nl, write('Computador Ganha'),nl.
ganhou(1):- nl, write('Jogador Ganha'),nl.

% Nim

agenteNim(Ei):- terminal(Ei), valor(Ei, V, 1), ganhou(V), escNim(Ei).
agenteNim(Ei):- minimax_decidir(Ei,Op, Ess), write('Agente joga - '), write(Op), escNim(Ess), jogadorNim(Ess).

jogadorNim(Ei):- terminal(Ei), valor(Ei, V, 2), ganhou(V), escNim(Ei).
jogadorNim(Ess):-lerJogadaNim(I,J), op1(Ess, tira(I,J), Ne), agenteNim(Ne).

lerJogadaNim(I,J):- write('linha = '), read(I), write('quantidade = '), read(J), nl.

% Writes

escLinha(0):-nl.
escLinha(N):- write(' |'), M is N-1, escLinha(M).

escNim(e(L1,L2,L3,L4)):- nl,write('1 - '), escLinha(L1), nl,write('2 - '), escLinha(L2),
    nl,write('3 - '), escLinha(L3), nl, write('4 - '),escLinha(L4),nl.
```

- h) Apresente uma tabela com o número de nós expandidos para diferentes estados do jogo (10 no mínimo) com os vários algoritmos.

ESTADOS DO JOGO	NÓS EXPANDIDOS
estado_inicial(e(1,1,1,2))	303
estado_inicial(e(1,2,3,2))	17825
estado_inicial(e(1,1,2,7))	251268
estado_inicial(e(1,0,2,5))	4541
estado_inicial(e(0,0,3,2))	122
estado_inicial(e(0,2,3,2))	2533
estado_inicial(e(1,2,1,1))	303
estado_inicial(e(1,0,1,1))	21
estado_inicial(e(1,3,5,1))	140165
estado_inicial(e(1,0,0,7))	877

Exercício 2

Breve descrição do jogo:

O jogo que escolhemos foi o wythoff's queen, este é jogado num tabuleiro de xadrez, com apenas uma Rainha. Ambos os jogadores movem a mesma rainha, alternadamente, esta pode mover-se apenas para a esquerda, baixo, e diagonal inferior esquerda. Ganha o jogador que colocar a rainha no quadrado (0,0) (Canto inferior esquerdo).

- a) Escolha uma estrutura de dados para representar os estados do jogo.

```
% num tabuleiro n*n, qualquer posição inicial pode ser escolhida  
estado_inicial(e(7,6)).
```

- b) Defina o predicado terminal(estado) que sucede quando um estado é terminal.

```
terminal(e(0,0)).
```

- c) Defina uma função de utilidade que para um estado terminal que deve retornar o valor do estado

```
valor(E,-1,P):- terminal(E), X is P mod 2, X=0,!.  
valor(E,1,_):- terminal(E).
```

- d) Implemente um agente inteligente que joga o jogo que escolheu usando várias estratégias, **minimax**, corte **alfa-beta** e corte em profundidade.

```
jogar(Jogo):- [Jogo], estado_inicial(E1),
    write('Inicio - '),nl,
    (Jogo=nim -> escNim(E1),agenteNim(E1); escWQueen(E1),agenteWQueen(E1)).

it(Ei, Ess):- minimax_decidir(Ei, Opf, Ess).

ganhou(-1):- nl, write('Computador Ganha'),nl.
ganhou(1):- nl, write('Jogador Ganha'),nl.

% Whythoff's Queen

agenteWQueen(Ei):- terminal(Ei), valor(Ei, V, 1), ganhou(V), escWQueen(Ei).
agenteWQueen(Ei):- minimax_decidir(Ei,Op, Ess), write('Agente joga - '), write(Op), escWQueen(Ess), jogadorWQueen(Ess).

jogadorWQueen(Ei):- terminal(Ei), valor(Ei, V, 2), ganhou(V), escWQueen(Ei).
jogadorWQueen(Ess):-lerJogadaWQueen(I,J), op1(Ess, move(I,J), Ne), agenteWQueen(Ne).

lerJogadaWQueen(I,J):- write('linha = '), read(I), write('coluna = '), read(J), nl.

% Writes

escWQueen(e(X,Y)):- nl, escWQueenAux(e(X,Y), e(0,7)).

escWQueenAux(_, e(0,-1)).
escWQueenAux(e(X,Y), e(7,Cy)):- ((X,Y)=(7,Cy) -> write(' R '); write(' - ')), nl, Ncy is Cy-1, escWQueenAux(e(X,Y), e(0, Ncy)).
escWQueenAux(e(X,Y), e(Cx,Cy)):- ((X,Y)=(Cx, Cy)-> write(' R '); write(' - ')), Ncx is Cx+1, escWQueenAux(e(X,Y), e(Ncx, Cy)).
```

- e) Apresente uma tabela com o número de nós expandidos para diferentes estados do jogo (10 no mínimo) com os vários algoritmos.

ESTADOS DO JOGO	NÓS EXPANDIDOS
e(7,6)	1458529
e(4,0)	23
e(4,3)	1687
e(7,5)	436205
e(7,2)	7383
e(3,3)	583
e(2,2)	67
e(3,2)	186
e(6,4)	44766
e(7,4)	122811