

# **3º Trabalho Prático**

## **Jogos de dois jogadores**

Licenciatura em Eng. Informática  
Inteligência Artificial



**André Baião 48092**  
**Gonçalo Barradas 48402**  
**Guilherme Grilo 48921**  
**Docente: Irene Pimenta**



1. Considere o jogo do 4 em linha. Joga-se num tabuleiro de 6 X 7, na vertical. Considere que tem peças brancas e peça pretas. O primeiro a jogar é o jogador que tem as peças brancas. As peças são largadas no início de uma coluna e fica na primeira linha vazia dessa coluna

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

A estrutura escolhida para representar um estado é composta por uma lista com 42 posições que representam o tabuleiro, e a cor do jogar que vai jogar.

```

1      e([ v,v,v,v,v,v,v,
2          v,v,v,v,v,v,v,
3          v,v,v,v,v,v,v,
4          v,v,v,v,v,v,v,
5          v,v,v,v,v,v,v,
6          v,v,v,v,v,v,v], w)

```

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

```

1      terminal(E):- dividir_em_linhas(E,SL),E = e(L,J),(linhas(e(SL),J);
2                  colunas(e(SL),J); diagonal(e(SL),J); cheio(L)).
3
4      cheio(L):- \+member(v,L).
5
6      linhas(e([]),_):- fail.
7      linhas(e([H|T]),X) :- linhas(H,X); linhas(e(T),X).
8      linhas([X,X,X,X|_],X):- X \= v.
9      linhas([_|T],X):- linhas(T,X).
10
11     colunas(e([]),_):- fail.
12     colunas(e([_,_,_,[]]),_):- fail.
13     colunas(e([A,B,C,D|T]),X) :- colunas(A,B,C,D,X); colunas(e([B,C,D|T
14                                     ]),X).
15     colunas([X|_],[X|_],[X|_],[X|_],X):- X \= v.
16     colunas([_|A],[_|B],[_|C],[_|D],X):- colunas(A,B,C,D,X).
17     colunas([],[],[],[],_):- fail.
18
19     diagonal(e([]),_):- fail.
20     diagonal(e([_,_,_,[]]),_):- fail.
21     diagonal(e([A,B,C,D|T]),X) :- diagonal(A,B,C,D,X); diagonal(e([B,C,
22                                     D|T]),X).
23     diagonal([X|_],[_,X|_],[_,_,X|_],[_,_,_,X|_],X):- X \= v.
24     diagonal([_,_,_,X|_],[_,_,X|_],[_,X|_],[X|_],X):- X \= v.
25     diagonal([_|A],[_|B],[_|C],[_|D],X):- diagonal(A,B,C,D,X).
26     diagonal([],[],[],[],_):- fail.
27
28     dividir_em_linhas(e(L,_), S) :- divide(L,S).
29     divide([],[]).
30     divide([V1,V2,V3,V4,V5,V6,V7|T],[[V1,V2,V3,V4,V5,V6,V7|T1]]):-
31         divide(T,T1).

```

- (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).

```

1      valor(e(L,_),0,P):- \+ member(v,L),!.
2      valor(E,V,P):-
3          terminal(E),
4          X is P mod 2,
5          ( X == 1, V = 1 ; X == 0, V = -1 ).

```

- (d) Use a implementação do minimax dada na aula prática para escolher a melhor jogada num estado. Teste a sua descrição do jogo com vários estados.

---

```

1      estado_inicial(e([ v,v,v,v,b,w,b,
2                        w,b,v,w,w,b,w,
3                        b,w,b,b,w,b,w,
4                        w,b,w,v,w,b,w,
5                        b,w,b,w,b,w,b,
6                        w,b,w,b,w,b,w], w))).
7
8      Resultado: 25

```

---

```

1      estado_inicial(e([ v,v,v,v,b,w,b,
2                        w,b,v,w,w,b,w,
3                        b,w,b,b,w,b,w,
4                        w,b,w,w,w,b,w,
5                        v,w,b,w,b,w,b,
6                        w,b,w,b,w,b,w], w))).
7
8      Resultado: 29

```

---

```

1      estado_inicial(e([ v,v,v,v,b,w,b,
2                        w,b,v,w,w,b,w,
3                        b,w,b,b,w,b,w,
4                        w,b,w,w,w,b,w,
5                        w,w,b,w,b,w,b,
6                        w,b,w,b,w,b,w], w))).
7
8      Resultado: 10

```

---

```

1      estado_inicial(e([ v,v,v,v,b,w,b,
2                        w,b,v,w,w,b,w,
3                        b,w,b,b,w,b,w,
4                        w,b,w,b,w,b,w,
5                        b,w,b,v,b,w,b,
6                        w,b,w,b,w,b,w], w))).
7
8      Resultado: 32

```

---

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

Após a tentativa de implementação do algoritmo Alfa-Beta em Prolog, deparamos nos com diversos problemas que dificultaram a implementação correta. Esses problemas podem estar relacionados a erros de sintaxe ou lógica na codificação do algoritmo, dificuldades em representar adequadamente o estado do jogo em Prolog ou outras questões específicas da linguagem.

Infelizmente sem a implementação correta do Alfa-beta não é possível comparar resultados.

2. Escolha um jogo de dois jogadores diferente do jogo do galo e do jogo do 31.

O jogo escolhido foi o jogo do Nim. O jogo do Nim é um jogo de estratégia matemática em que dois jogadores jogam ha vez para remover objetos de pilhas. O objetivo é evitar ficar com a última peça, pois quem ficar com a última peça perde. É um jogo que envolve pensamento estratégico, raciocínio lógico e análise de padrões.

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

A estrutura escolhida é uma estrutura simples que representa os 4 montes e o numero de peças em cada monte.

---

```

1      e(1, 2, 2, 5)

```

---



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

---

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

---

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

---

```
1      valor(E, -1, P):- terminal(E),
2                                R is P mod 2,
3                                R = 1.
4
5      valor(E, 1, P):- terminal(E),
6                                R is P mod 2,
7                                R=0.
```

---

- (d) (e)

Derivado à inúmera presença de bugs e problemas como o desenvolvimento com a linguagem declarativa Prolog, não conseguimos implementar o algoritmo Alfa-Beta, nem o agente inteligente.