3º Trabalho de Inteligência Artificial

2018/2019



Jogos de dois jogadores – Jogos com informação completa determinísticos

Docente:  
Irene Pimenta Rodrigues

Realizado por:

Daniel Serrano – 35087

Miguel Serrano - 34149

09 de abril de 2019

Conteúdo

[Introdução: 3](#_Toc5851214)

[1º Exercício: 4](#_Toc5851215)

[1.a) 4](#_Toc5851216)

[1.b) 4](#_Toc5851217)

[1.c) 5](#_Toc5851218)

[1.d) 5](#_Toc5851219)

[1.e) 6](#_Toc5851220)

[1.f) 6](#_Toc5851221)

[1.g) 7](#_Toc5851222)

[1.h) 7](#_Toc5851223)

[2º Exercício: 8](#_Toc5851224)

[2.a) 8](#_Toc5851225)

[2.b) 8](#_Toc5851226)

[2.c) 9](#_Toc5851227)

[2.d) 9](#_Toc5851228)

[2.e) 10](#_Toc5851229)

[Conclusão: 11](#_Toc5851230)

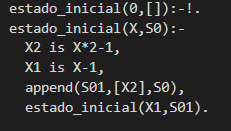
Introdução:

Neste trabalho, no âmbito da cadeira de Inteligência Artificial iremos resolver o jogo do nim, jogo proposto pela professora, e o jogo do galo, jogo escolhido por nós visto termos começado a implementá-lo numa aula prática. Para resolução destes jogos por parte do CPU iremos utilizar os algoritmos de minmax e alfabeta.

1º Exercício:

1.a)

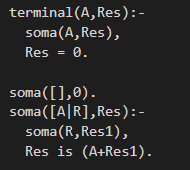
A estrutura de dados escolhida por nós, para representar os estados do jogo foi a seguinte:



Estrutura esta que recebe como input um valor X que equivale ao número de colunas e a partir desse valor coluna 2\*X -1 peças em jogo por cada coluna, e diminui 1 valor esse mesmo X, de modo a que a última coluna a colocar peças fique apenas com uma peça. Exemplo: Se X=3 o estado será do tipo: [1,3,5].

1.b)

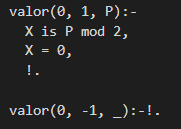
Para determinar o estado terminal utilizámos o predicado:



Em que é calculada a soma das peças em todas as colunas e o estado será então terminal, e de acordo com as regras do jogo, quando o número de peças for 0.

1.c)

Para definir a função utilidade utilizámos o seguinte predicado:



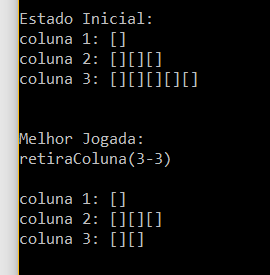
Função que ao atingir um estado terminal verifica se a profundidade é par ou não. Caso seja par, a função retorna o valor 1 o que significa que o jogador principal ganhou. Caso seja ímpar, retorna o valor -1 e o jogador vitorioso é o adversário. De reparar que não temos um caso de empate pois isso é impossível neste jogo.

1.d)

Para este ponto foi necessário então implementar o algoritmo de minimax. Para tal, utilizámos o algoritmo disponibilizado pela professora, mas realizámos algumas alterações de modo a que se adaptasse ao nosso jogo.

Para utilizar este algoritmo para escolher a melhor jogada teremos de seguir os seguintes passos:

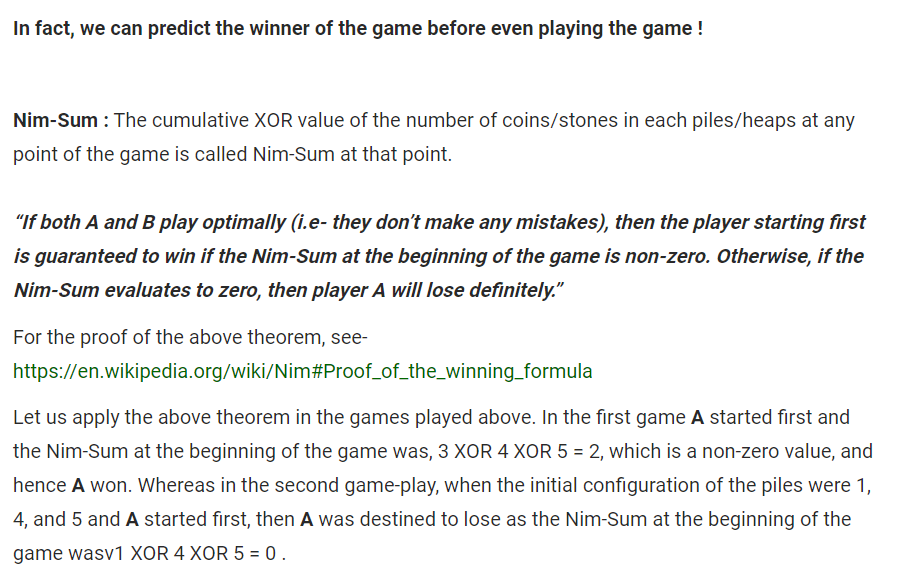
1. ?- [main].
2. ?- main.
3. Introduza o número de linhas: 3. (à escolhe do user)
4. Escolha Algoritmo: \n 1. Minimax \n 2. AlfaBeta \n : 1.
5. Escolha Modo: \n 1. Melhor Jogada \n 2. Homem vs CPU \n 3. CPU vs CPU \n : 1.
6. Insira a Profundidade: 10. (à escolha do user)
7. Output do programa:



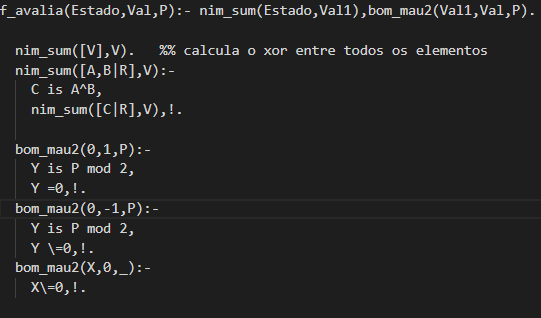
1.e)

1.f)

Para a função de avaliação consultámos um site ([link](https://www.geeksforgeeks.org/combinatorial-game-theory-set-2-game-nim/)) em que conseguimos perceber o que a esta função devia testar:

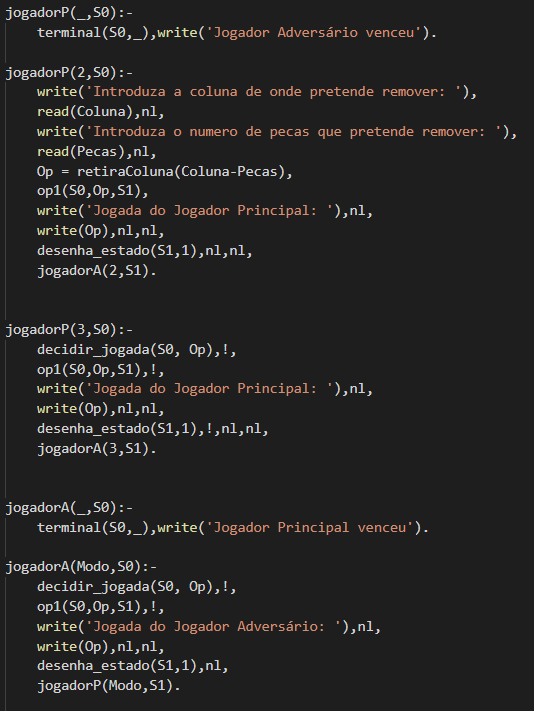


Portanto, o que esta função irá fazer é fazer a operação xor entre o número de elementos de cada coluna, e, caso o resultado seja 0 significa que estamos num estado de vantagem, ou seja, Se estivermos numa profundidade par, e o valor da função nim-sum for 0 significa que o jogador principal está em vantagem, no entanto também estará em vantagem se numa profundidade impar o valor da função nim-sum não der 0. Se nestes casos em que quem está em vantagem é o jogador principal a função avaliação retornará o valor 1, caso contrário retornará o valor -1.



1.g)

Para a implementação de um agente inteligente optámos por permitir que o utilizador pudesse jogar contra um PC ou então visualizar o jogo entro 2 PC’s. Este agente pode ser mais difícil se a profundidade limite definida pelo utilizador no início de cada jogo for maior.  
De referir que num jogo entre dois PC’s irá ganhar aquele que primeiro atingir um estado de vantagem.



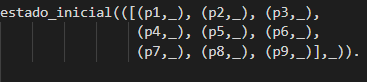
1.h)

2º Exercício:

Para o 2º exercício era pedido um jogo de dois jogadores, e como já tínhamos iniciado a programação do jogo do galo num das aulas práticas decidimos continuá-lo neste pronto.

2.a)

A estrutura de dados escolhida por nós, para representar os estados do jogo foi a seguinte:



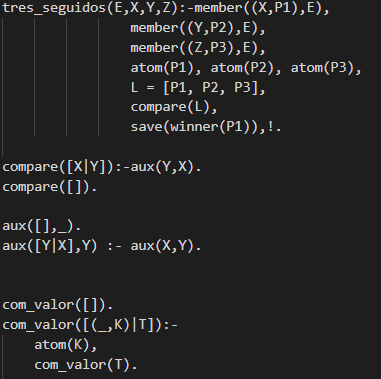
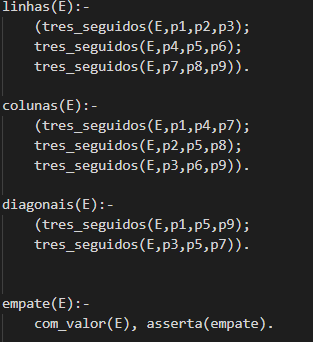
Estrutura esta que como todos os jogos do galo, o estado inicial não tem nenhuma posição preenchida e as posições são representadas por pPos.

2.b)

Para determinar o estado terminal utilizámos o predicado:

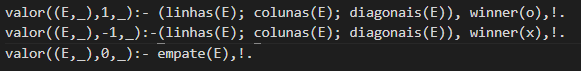


Em que verifica se há alguma sequencia de símbolos iguais tanto nas linhas, como nas colunas, como nas diagonais, e caso isso não se verifique, irá também verificar se ocorreu algum empate. Este predicado é auxiliado pelas seguintes funções:



2.c)

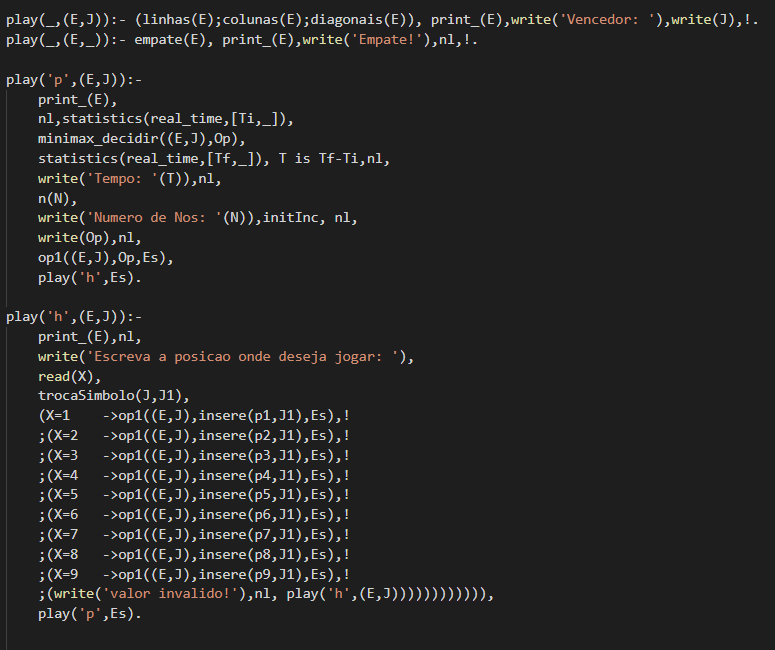
Para definir a função utilidade utilizámos o seguinte predicado:



Função verifica a profundidade na arvora de pesquisa e os casos em que o estado é terminal, com a exceção de empate. Os valores devolvidos pela mesma podem ser 1, 0 ou -1 sendo que 0 representa empate, 1 ganha e -1 perde.

2.d)

Para este jogo apenas optámos por não colocar nenhum método de visualizar 2 PC’s a jogar um contra o outro, mas apenas um utilizador contra um PC. A função que nos permite isto é a seguinte:



2.e)

|  |  |
| --- | --- |
| Minimax | Estado |
| 59704 | Posição inicial: 1 |
| 63904 | Posição inicial: 2 |
| 59704 | Posição inicial: 3 |
| 63904 | Posição inicial: 4 |
| 55504 | Posição inicial: 5 |
| 63904 | Posição inicial: 6 |
| 59704 | Posição inicial: 7 |
| 63904 | Posição inicial: 8 |
| 59704 | Posição inicial: 9 |
| 934 | Posição inicial: 1 + 2º posição: 1 |
| 46 | Posição inicial: 1 + 2º posição: 2 + 3º Posição: 7 |
| 4 | Posição inicial: 1 + 2º posição: 2 + 3º Posição: 7 + 4ª Posição: 6 |

Conclusão:

Com a realização deste trabalho ficámos a ter mais conhecimento sobre os tipos de jogos com dois jogadores, e apesar de não termos conseguido implementar o algoritmo de alfabeta conseguimos calcular que seria muito mais eficiente que o minimax, visto este ter sido bastante lento quando se tratar de jogos muito exigentes, como se verificou no jogo do galo.