



# Relatório do 3º Trabalho Prático de Inteligência Artificial

Henrique Raposo nº33101

José Serra nº 33289

10 Abril, 2019

## 1 Introdução

Como enunciado, tratámos o jogo do Nim como um jogo de dois jogadores com informação completa e determinístico, escolhemos também tratar da mesma maneira o jogo do galo e representar ambos em Prolog, utilizando também os algoritmos de MinMax e Alfa-Beta, dados nas aulas teóricas da disciplina, de maneira a responder às várias alíneas enunciadas.

## 2 Jogo do Nim

### 2.1 Alínea a): Estrutura de Dados que representa o problema

A maneira utilizada para representar o problema em questão foi utilizar uma lista com o número de peças que cada linha contém, como o número de peças em cada linha difere de acordo com a profundidade utilizá-mos a expressão  $2 * n + 1$  para obter o correspondente número.

### 2.2 Alínea b) Predicado terminal

O predicado terminal é o que decide quando o jogo termina, que neste caso é quando todas as peças são utilizadas ou seja quando o número das peças em cada linha for 0.

```
terminal(A,Res):-  
    soma(A,Res),  
    Res = 0.  
  
soma([],0).  
soma([A|R],Res):-  
    soma(R,Res1),  
    Res is (A+Res1).
```

### 2.3 Alinea c) Função utilidade

A função utilidade serve para determinar, quando o jogo termina o jogador que ganhou através da profundidade, visto ser um jogo com dois jogadores, se a profundidade for par, o jogador par é que perde logo a função vai retornar 1, se ocorrer o contrário a função vai retornar -1.

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

### 2.4 Alinea d) Pesquisa Minmax

A pesquisa Minimax é método dado nas aulas teóricas para minimizar a possível perda máxima. Para este trabalho utilizámos o algoritmo dado nas aulas porém com algumas alterações para o problema em questão.

### 2.5 Alinea e) Pesquisa Alfa-Beta

O algoritmo Alfa-Beta consiste numa variação do algoritmo minimax , que visa reduzir o número de nós que são avaliados criando assim um algoritmo mais eficiente.

### 2.6 Alinea f) Função avaliação

De acordo com o enunciado foi definida uma função avaliação que impede o programa de aprofundar muito a pesquisa do estado terminal , impedindo assim que este programa rebente e tornando-o mais rápido.

### 2.7 Alinea g) Agente Inteligente

Foi implementado um agente inteligente , que ao escolhermos o algoritmo calcula e joga a melhor jogada consoante a profundidade dada. Os algoritmos possíveis de escolher s o o MinMax e o AlfaBeta , é também para cada algoritmo, possível jogar um humano contra este, ou o algoritmo escolhido contra ele próprio.

## 3 Jogo do Galo

### 3.1 Alinea a): Estrutura de Dados que representa o problema

A maneira utilizada para representar o problema em questão foi utilizar uma lista de tuplos com as diferentes coordenadas e o respectivo valor da jogada.

```
estado_inicial(((p(1,1), _), (p(1,2), _), (p(1,3), _),  
                (p(2,1), _), (p(2,2), _), (p(2,3), _),  
                (p(3,1), _), (p(3,2), _), (p(3,3), _)] , _)).
```

### 3.2 Alinea b) Predicado terminal

O predicado terminal decide quando é que o jogo termina, que neste caso, é quando uma linha, coluna ou diagonal da mesma peça ocorre, caso o tabuleiro esteja cheio e não seja possível fazer mais jogadas a função deteta o empate.

```
terminal((E, _)):-  
    linhas(E);colunas(E);diagonais(E);empate(E).
```

### 3.3 Alinea c) Função utilidade

A função de utilidade verifica a profundidade na árvore de pesquisa e os casos em que o estado é terminal, com a excepção do empate. Os valores devolvidos pela mesma podem ser 1, 0 ou -1 sendo que 0 representa empate, 1 ganha e -1 perde.

```
valor((E, _), 1, _):- (linhas(E);colunas(E);diagonais(E)), ganhador(o), !.  
valor((E, _), -1, _):- (linhas(E);colunas(E);diagonais(E)), ganhador(x), !.  
valor((E, _), 0, _):- empate(E), !.
```

### 3.4 Alineas d) e e) Implementação dos algoritmos e tabela de valores de comparação

Neste trabalho foi implementado apenas o algoritmo minmax. Na imagem que segue, consta a média de 5 observações para cada jogada calculada pelo algoritmo:

nº de jogadas efectuadas	tempo (ms)	nós expandidos
0	33670,5	549945
1	3416	57604
2	476	7781
3	70	1053