

FICHA DE LABORATÓRIO Nº12

O JOGO DO GALO COM O ALGORITMO ALFA BETA

Inteligência Artificial - Escola Superior de Tecnologia de Setúbal

2015 / 2016

Objetivos da ficha

- Implementar o algoritmo AlfaBeta
- Editar um programa para jogar o jogo do Galo com o algoritmo AlfaBeta.

O laboratório tem como objetivo implementar o jogo do Galo em LISP, utilizando o algoritmo AlfaBeta para decidir as jogadas do computador. Com base no conhecimento do jogo adquirido através dos exercícios da ficha de laboratório nº 11, irão desenvolver um conjunto de funções que permite jogar o jogo do galo com o algoritmo AlfaBeta. Para desenvolver o programa, irão utilizar o ficheiro **laboratorio12.lisp**, disponível no Moodle, que já contém uma versão funcional do jogo, na qual o computador joga recorrendo a uma estratégia aleatória. Neste laboratório, deverão desenvolver apenas as funções para implementar o algoritmo AlfaBeta e editar a função **escolhe-melhor-jogada** para permitir que o computador passa a usar o AlfaBeta como estratégia de jogo.

1 O jogo do galo

O jogo do galo é jogado num tabuleiro de 9 casas entre dois jogadores.

1.1 Regras

1. O tabuleiro é constituído por uma matriz de três linhas por três colunas.
2. Cada jogador joga com um símbolo diferente. Os símbolos usados pelos jogadores são [O] para o jogador humano e [X] para o PC.
3. Os jogadores jogam alternadamente, colocando a respetiva marcação no tabuleiro, numa casa que esteja vazia.

1		O				X	
2				O		X	
3						X	

Figura 1 - Exemplo de sequência do jogo do galo, ganho pelo PC

2 O programa do jogo

Considere que a representação de um tabuleiro, em Lisp, se faz através de uma lista de 9 elementos, na qual os primeiros 3 representam a linha 1, os elementos 4 a 6 representam a linha 2 e os elementos 7 a 9 representam a linha 3. As peças do jogador MAX (O PC) são representadas pelo valor [10] e as do jogador MIN (O humano) pelo valor [1]. As casas vazias são representadas pelo valor [0].

1	(1 0 10
2	0 1 10
3	0 0 10)

Figura 2 - Representação em Lisp do tabuleiro do jogo apresentado na figura 1

O tipo abstrato que permite representar um nó da árvore do jogo consiste numa lista composta por dois elementos:

(tabuleiro profundidade)

Nesta lista,

- o primeiro elemento, estado, representa o tabuleiro do jogo;
- o segundo representa a profundidade do nó na árvore do jogo.

O nó inicial é sempre do tipo [MAX].

2.1 Exercícios

No ficheiro dado, existe um conjunto de funções para o programa. Análise as DocStings de cada função para perceber o que as funções fazem.

1. Defina uma função **avaliar-no** que recebe um tabuleiro e um jogador (MAX | MIN) e que retorna:

- se o nó for terminal:
 - o valor [100] em caso de vitória do jogador MAX,
 - o valor [-100] em caso de vitória do jogador MIN
 - o valor [0] em caso de empate.
- se não for terminal e se a casa central do tabuleiro for preenchida:
 - o valor [-50] se o jogador MIN tiver uma peça no centro do tabuleiro
 - o valor [50] se o jogador MAX tiver uma peça no centro do tabuleiro

,

2. Defina uma função **alfabeta** que permite determinar a jogada a fazer pelo computador. A função recebe um nó e um valor inteiro que corresponde a profundidade limite para a expansão da árvore do jogo. A função retorna o valor da função utilidade da melhor jogada. O tabuleiro que corresponde a melhor jogada é guardado na variável global *jogada* que é atualizada pelo algoritmo cada vez que o valor de beta é atualizado. A função **alfabeta** é invocada na função **escolhe-melhor-jogada** no código fornecido, com o valor 4 para a profundidade limite.
3. Redefina a função **avaliar-no** por forma a adicionar uma tática atacante ou defensiva à função, por forma a melhorar o desempenho do algoritmo alfabeta.