

Inteligência Artificial

LE4 – Lista de Exercícios 4

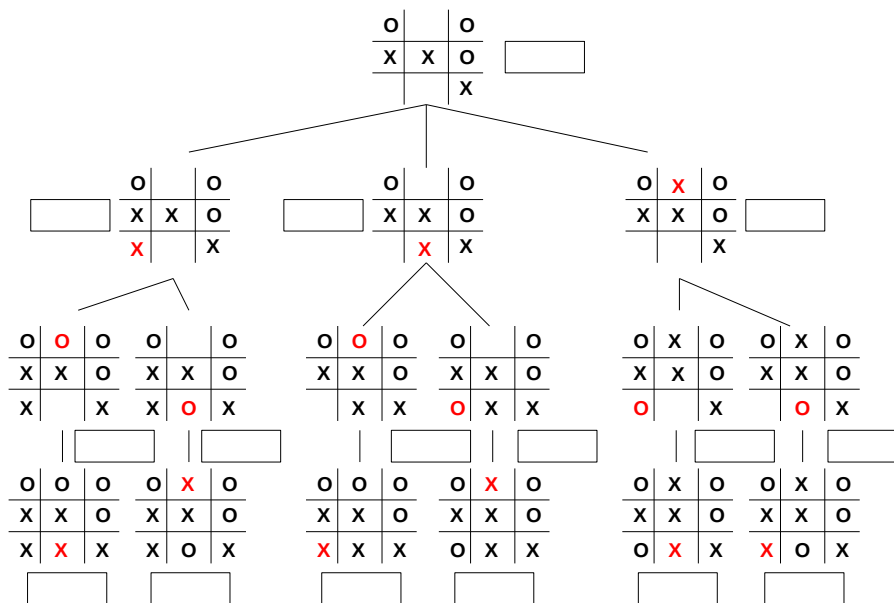
Assunto: Preparação para a Prova Final.

I - Observações:

1. Não é preciso entregar a lista de exercícios.

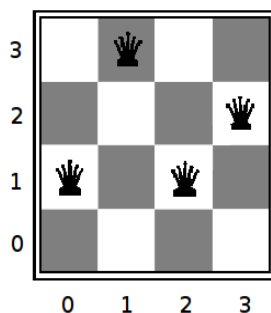
II – Questões

1. Em alguns problemas de busca, principalmente em jogos, pode ser impossível atingir os nós terminais em tempo hábil para se efetuar uma jogada. Neste caso, emprega-se uma heurística para definir o valor para cada um dos nós da árvore. Assim, é possível ter uma “ideia” do valor da jogada sem ser necessário atingir os nós terminais. O valor atribuído a cada nó da árvore deve ter uma forte relação com a probabilidade do jogador conseguir uma vitória executando aquela jogada. Para a árvore de busca a seguir, considerando que o algoritmo *minimax* está sendo empregado com *busca em profundidade iterativa*, pede-se:

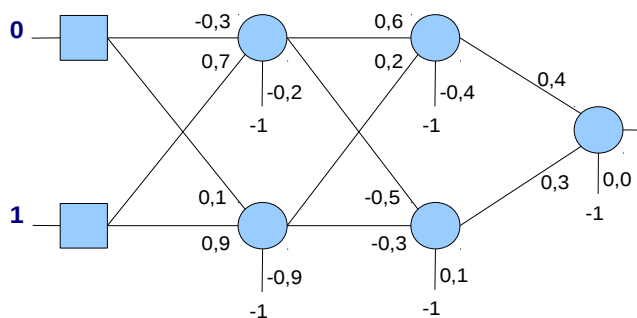


- a) Determinar uma heurística simples para calcular o valor de cada nó da árvore. Explique detalhadamente como um algoritmo poderia calcular o valor do nó utilizando esta heurística. Calcule o valor de cada nó da árvore do exercício e coloque este valor nos retângulos ao lado de cada nó. OBS: Sua heurística somente pode utilizar dados do próprio nó para calcular seu valor.
 - b) Qual a jogada um agente utilizando *minimax* executaria, a partir do nó raiz, se a profundidade máxima da busca usando *profundidade iterativa* fosse 2? E se fosse 3?
 - c) Se a sua heurística fosse empregada nos nós terminais da árvore de busca, ela permitiria que o agente fizesse a escolha correta da jogada?
2. Execute manualmente o algoritmo *A** para o problema de chegar à Bucareste a partir de Vaslui usando a heurística de distância em linha direta. Mostre a sequência de nós que o algoritmo vai considerar a cada iteração e o valor de f , g , e h para cada um destes nós. O mapa da Romênia e o valor das distâncias em linha reta para as cidades até Bucareste estão nas transparências vistas em sala de aula.

3. Execute manualmente o algoritmo *Hill Climbing* para determinar o posicionamento final das rainhas, considerando o estado inicial representado na figura abaixo. Para tal, considere que os únicos vizinhos deste estado são aqueles em que a rainha da coluna 2 se movimenta para as linhas 0, 2 e 3. O valor de cada um dos estados é calculado utilizando o número de colisões. Todos os vizinhos devem ser testados e aquele com menor valor da heurística é escolhido como sucessor. Deixe indicado qual o valor de cada um dos 3 vizinhos deste estado e qual a posição final da rainha da coluna 2.



4. Crie regras em prolog para representar as relações de parentesco *avô*, *sobrinho(a)* e *parente*. Para isto, considere que sua base de dados contém apenas fatos utilizando as relações *genitor*, *masculino* e *feminino*. Considere que a relação binária *parente* indica que duas pessoas têm um ascendente em comum. OBS: Utilize os recursos do laboratório 2 para testar suas regras.
5. Considere o problema de fazer com que um Perceptron Simples aproxime a função NAND.
- Quais os valores dos pesos das entradas e do peso do *bias* (theta) para que o perceptron simule corretamente esta função?
 - Execute manualmente o algoritmo de aprendizagem do Perceptron para as duas primeiras linhas da tabela verdade da função NAND. Considere que todos os pesos (inclusive o *theta*) iniciam-se com valor 0. Também considere que as duas primeiras linhas da tabela verdade são: $\{x_1 = 0, x_2 = 0\}$ e $\{x_1 = 0, x_2 = 1\}$.
6. É provado que uma MLP com 2 ou mais camadas ocultas pode aproximar qualquer função. Explique por que, apesar desta afirmação ser verdadeira, nem sempre é possível encontrar soluções de algumas instâncias de problemas utilizando MLPs.
7. Calcule o valor da saída da MLP abaixo, considerando a função de ativação $f(x)$.



$$f(x) = \begin{cases} -1 & \text{se } x < -1 \\ x & \text{se } -1 \leq x \leq 1 \\ 1 & \text{se } x > 1 \end{cases}$$

BOM TRABALHO!