|  |  |
| --- | --- |
| Ícone  Descrição gerada automaticamente | PONTIFÍCIA UNIVERSIDADE CATÓLICA DO PARANÁ  Escola Politécnica    Curso: Ciência da Computação  Disciplina: Inteligência Artificial |

 Avaliação INDIVIDUAL Somativa – 18/maio/2021 – Teoria de Jogos

INSTRUÇÕES:

a) Leia as questões atentamente antes de responder;

b) Todas as respostas deverão estar nesta folha;

c) Enviar (INDIVIDUALMENTE) o arquivo com as respostas usando no ***Blackboard*** o item ***Mensagens/Criar Mensagem/Destinatário: JULIO CESAR NIEVOLA . (Professor)*** até as 23h00 do dia 18/maio/2021 (HOJE).

Nome: \_\_\_\_\_Gustavo Hammerschmidt.\_\_\_\_\_\_\_\_\_\_

1) (*1,0 ponto*) Como deve ser alterado o algoritmo alfa-beta para levar em conta um fator aleatório em um jogo (tal como o sorteio de um dado)?

Deve-se criar uma ordem de poda que se ajusta conforme o fator sorte indicar; para isso, eu faria uma rotina que define precedências entre jogadas, por exemplo, logo após o lançamento de um dado, o algoritmo deve checar os galhos da árvore em que este fator sorte surte mais efeito e aumenta a probabilidade de vitória – para isso, basta organizar a ordem de avaliação dos galhos pelo o algoritmo de poda, começando com o que tem mais chances, assim os galhos com menores chances serão podados a cada jogada.

2) (*1,0 ponto*) Comparando o algoritmo MiniMax com o algoritmo Alfa-Beta, podemos afirmar:

I – Ambos percorrem a árvore de possibilidades seguindo o mesmo caminho.

II – O algoritmo Alfa-Beta avalia um número de estados menor ou igual ao número de estados avaliado pelo algoritmo MiniMax.

III – O algoritmo Alfa-Beta utiliza uma função heurística para avaliar os estados, enquanto o algoritmo MiniMax usa uma função de utilidade.

IV – A complexidade de implementação do algoritmo Alfa-Beta é maior que a do algoritmo MiniMax.

V – Em ambos os algoritmos o usuário determina a profundidade da poda da árvore a ser utilizada.

São corretas, apenas, as afirmações:

a) I

b) V

c) I e V

d) III e IV

e) II, III e IV.

Resposta: São verdadeiras as afirmações: II, III e IV, alternativa e.

3) (*2,5 pontos*) Considere um jogo do tipo 8-puzzle, cujo objetivo é conduzir o tabuleiro esquematizado na figura abaixo para o seguinte estado final.

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 3 |
| 8 |  | 4 |
| 7 | 6 | 5 |

Figura 1 – Estado final desejado.

Considere, ainda, que, em determinado instante do jogo, se tenha o estado E0 a seguir.

|  |  |  |
| --- | --- | --- |
| 3 | 4 | 6 |
| 5 | 8 |  |
| 2 | 1 | 7 |

Figura 2 – Estado E0.

Pelas regras desse jogo, sabe-se que os próximos estados possíveis são os estados E1, E2 e E3 mostrados abaixo.

|  |  |  |
| --- | --- | --- |
| 3 | 4 | 6 |
| 5 |  | 8 |
| 2 | 1 | 7 |

Figura 3 – Estado E1.

|  |  |  |
| --- | --- | --- |
| 3 | 4 | 6 |
| 5 | 8 | 7 |
| 2 | 1 |  |

Figura 4 – Estado E2.

|  |  |  |
| --- | --- | --- |
| 3 | 4 |  |
| 5 | 8 | 6 |
| 2 | 1 | 7 |

Figura 5 – Estado E3.

Considere uma função heurística ***h*** embasada na soma das distâncias das peças em relação ao estado final desejado, em que a distância ***d*** que uma peça **p** está da posição final é dada pela soma do número de linhas com o número de colunas que a separam da posição final desejada. Por exemplo, em E1, ***d***(1) = 2(linhas) + 1(coluna) = 3. A partir dessas informações analise as asserções a seguir.

**Asserções**:

**A1:**Utilizando-se um algoritmo de busca gulosa pela melhor escolha que utiliza a função h, o próximo estado no desenvolvimento do jogo a partir do estado E0 tem de ser E3.

**PORQUE**

**A2:**Dos três estados E1, E2 e E3 possíveis, o estado com menor soma das distâncias entre a posição atual das peças e a posição final é o estado E3.

Assinale a opção correta a respeito dessas asserções:

A) As duas asserções são proposições verdadeiras, e a segunda é uma justificativa correta da primeira.

B) As duas asserções são proposições verdadeiras, e a segunda não é uma justificativa correta da primeira.

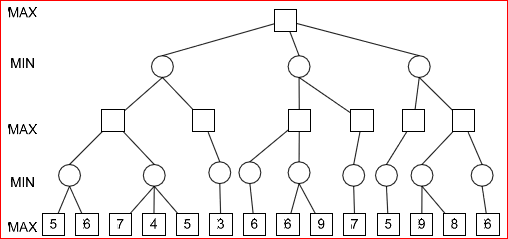
C) A primeira asserção é uma proposição verdadeira, e a segunda é uma proposição falsa.

D) A primeira asserção é uma proposição falsa, e a segunda é uma proposição verdadeira.

E) As duas asserções são proposições falsas.

Resposta: alternativa a.

4) (*3,5 pontos*) Considere a árvore minimax abaixo, representando um jogo onde queremos maximizar o valor da função de avaliação estática:

****

Assinale a alternativa que apresenta a quantidade de nós que não serão visitados em uma busca da melhor jogada se a estratégia de poda (algoritmo) ***alfa-beta*** for utilizada.

A) 5

B) 8

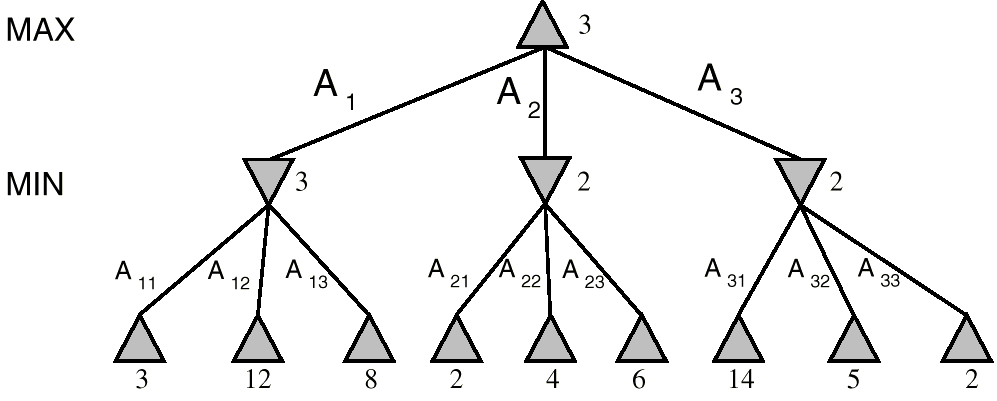
C) 9

D) 10

E) 11

Resposta: 8 nós não serão percorridos, alternativa b.

5) (*2,0 pontos*) Usando o algoritmo MiniMax, qual seria a sequência de ações sendo testados para o problema com espaço de estados indicado a seguir?



Ação 1: árvore pega o máximo dos três mínimos (primeiros nós a partir da raiz) [eu estou usando recursividade para obter os dados nesse algoritmo].

Ação 1: comando -> max(min(nó 1), min(nó 2), min(nó 3))

Ação 2: pegar o mínimo para o primeiro nó.

Ação 2: comando -> min (3, 12, 8) retornando 3.

Ação 3: pegar o mínimo para o segundo nó.

Ação 3: comando -> min (2, 4, 6) retornando 2.

Ação 4: pegar o mínimo para o terceiro nó.

Ação 4: comando -> min (14, 5, 2) retornando 2.

Ação 5: executaria o máximo para os valores obtidos.

Ação 5: comando -> max (3, 2, 2) retornando 3.

Observação: o algoritmo testaria para cada folha se ela é ou não uma folha, portanto, antes de cada ação com a instrução <comando>, teria se a chamada da função min ou max para cada ponteiro folha da árvores que retornaria ou uma função com ponteiros para avaliação-eager ou um valor inteiro para continuar a resolução. Neste cenário, o valor retornado nos nós folhas já é um inteiro para simplificação. Lembre-se que o teste é folha ou pai ocorre para cada nó durante a execução de todo o algoritmo.