

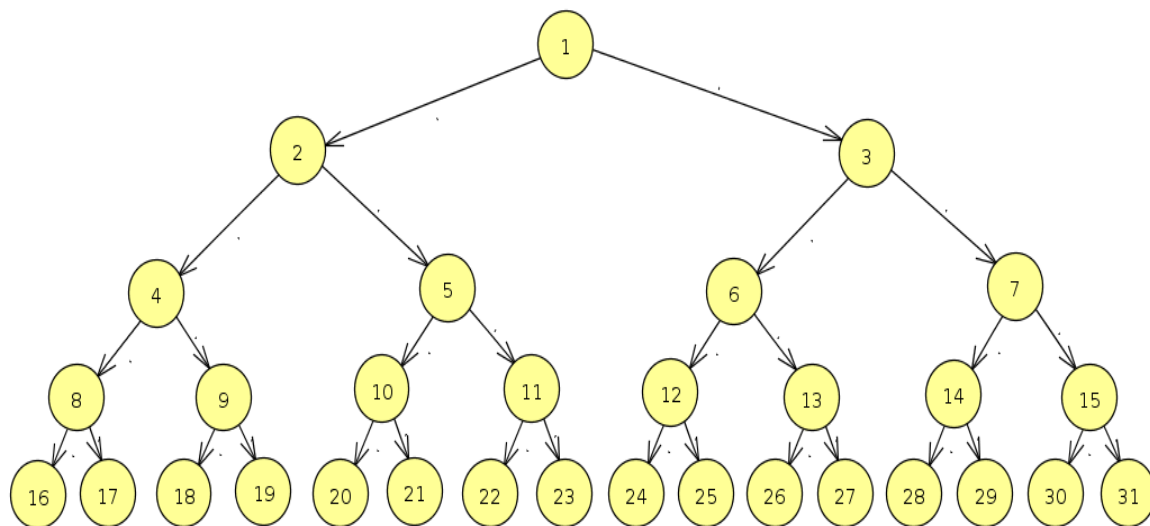
Inteligência Artificial

Thiago Henrique Leite da Silva, RA: 139920

AULA4: Exercício teórico - Busca sem informação (PT2)

1) (0,8) Considere um espaço de estados onde o estado inicial é o número 1 e cada estado k tem dois sucessores: números $2k$ e $2k+1$.

a. Represente a porção do espaço de estados para os estados de 1 a 15.



Para este espaço de estados, teremos sempre $2^n + 1$ estados, como nosso n no exemplo é 15, temos 31 estados distribuídos conforme a imagem acima, onde um estado K , nos leva a seu dobro e o sucessor dele.

b. Suponha que o estado objetivo seja 11. Liste a ordem em que os nós serão visitados pela busca em largura, busca em profundidade limitada com o limite 3 e busca de aprofundamento iterativo.

Busca em Largura:

- Enfileira o vértice inicial, 01
- Começa a verificação, desenfileira 01
- Verifica se seus filhos foram visitados, como não foram, enfileira-os.
- 02 e 03 são enfileirados.
- Desenfileira 02 e verifica seus filhos, 04 e 05
- Enfileira 04 e 05
- Desenfileira 03 e verifica seus filhos

- Enfileira 06 e 07
- Desenfileira 04 e verifica seus filhos
- Enfileira 08 e 09
- Desenfileira 05 e verifica seus filhos
- Enfileiramos o 10
- Chegamos no 11, fim da busca.

Portanto, a busca em largura neste caso percorre estes vértices:

01, 02, 03, 04, 05, 06, 07, 08, 09 e 10 até chegar no 11.

Busca em Profundidade Limitada:

Adotaremos o limite sendo 3.

- Começamos visitando 01 e inserindo ele na lista.
- Buscamos seus adjacentes, 02 e 03.
- Visitamos em profundidade 02 e inserimos na lista.
- Buscamos seus adjacentes, 04 e 05.
- Visitamos em profundidade 04 e inserimos na lista.
- Buscamos seus adjacentes 08 e 09.
- Chegamos no limite 3, não visitamos mais em profundidade.
- Fazemos o backtracking e vamos para 09.
- Visitamos em profundidade 09 e inserimos na lista.
- Chegamos no limite 3, não visitamos mais em profundidade.
- Fazemos o backtracking e vamos para 05.
- Visitamos em profundidade 10 e inserimos na lista.
- Chegamos no limite 3, não visitamos mais em profundidade.
- Fazemos o backtracking e vamos para 11.
- Chegamos no 11, busca encerrada.

Sequência de vértices visitados: 01, 02, 04, 08, 09, 05, 10 e 11

Busca de aprofundamento iterativo:

- Começamos com limite 0
- Visitamos 01, não achamos a solução.
- Reiniciamos com limite 1
- Visitamos 01 e buscamos seus adjacentes 02 e 03
- Visitamos 02, não é a solução e chegamos no limite.
- Visitamos 03, não é a solução e chegamos no limite.
- Reiniciamos com limite 2
- Visitamos 01 e buscamos seus adjacentes 02 e 03
- Visitamos 02 e seus adjacentes 04 e 05
- Visitamos 04, não é a solução e chegamos no limite.
- Visitamos 05, não é a solução e chegamos no limite.
- Visitamos 03 e seus adjacentes 06 e 07

- Visitamos 06, não é a solução e chegamos no limite.
- Visitamos 07, não é a solução e chegamos no limite.
- Reiniciamos com limite 3
- Visitamos 01, não achamos a solução.
- Buscamos seus adjacentes, 02 e 03.
- Visitamos em profundidade 02 e inserimos na lista.
- Buscamos seus adjacentes, 04 e 05.
- Visitamos em profundidade 04 e inserimos na lista.
- Buscamos seus adjacentes 08 e 09.
- Chegamos no limite 3, não visitamos mais em profundidade.
- Fazemos o backtracking e vamos para 09.
- Visitamos em profundidade 09 e inserimos na lista.
- Chegamos no limite 3, não visitamos mais em profundidade.
- Fazemos o backtracking e vamos para 05.
- Visitamos em profundidade 10 e inserimos na lista.
- Chegamos no limite 3, não visitamos mais em profundidade.
- Fazemos o backtracking e vamos para 11.
- Chegamos no 11, busca encerrada.

Portanto, para chegar no destino, passamos pelos seguintes vértices:

01,

01, 02, 03,

01, 02, 04, 05, 03, 06, 07,

01, 02, 04, 08, 09, 05, 10 e 11

c. Como a busca bidirecional funcionaria nesse problema? Qual é o fator de ramificação em cada direção?

A busca bidirecional nos permitiria realizar duas buscas simultâneas, uma partindo do vértice inicial, e uma partindo do vértice final, para que quando algum nó dessas duas árvores coincidisse, nós teríamos o caminho para chegar do vértice de início até vértice destino. Esta busca nos permite implementarmos da maneira que quisermos essas “sub-buscas”.

O fator de ramificação para a busca partindo do início seria 02, pois cada nó só possui dois adjacentes. Já o fator para a busca partindo do fim seria 03, pois podemos ir para os dois nós mais abaixo, e também o nó mais acima, algo que não acontece na busca do início, já que o nó raiz, do qual partimos, não tem nó pai.

O índice ao lado da instrução da busca representa a ordem que elas ocorrerão:

Partindo do estado inicial

- (1) Começamos visitando 01 e expandimos 02 e 03.
- (3) Visitamos 02 e expandimos 04 e 05.
- (5) Visitamos 04 e expandimos 08 e 09.

Partindo do estado final

- (2) Visitamos 11 e expandimos 05, 22 e 23.
- (4) Visitamos 05 e expandimos 02 e 10.
- (6) Visitamos 02, que está na árvore que partiu do início, fim da busca.

Concluindo, passaríamos pelos vértices:

01, 11, 02, 05 e 04 até que a busca se encerrasse, chegando ao caminho 01 -> 02 -> 05 -> 11.

As buscas escolhidas foram busca de aprofundamento limitado em 3 para ambas as árvores.

d. Chame a ação que vai de k para $2k$ de Esquerda e a ação de k que vai para $2k + 1$ de Direita. É possível encontrar um algoritmo que devolva a solução desse problema absolutamente sem nenhuma busca?

2) (0,2) O problema de missionários e canibais é normalmente enunciado como a seguir. Três missionários e três canibais estão em um lado de um rio, juntamente com um barco que pode levar uma ou duas pessoas. Descubra um meio de fazer todos atravessarem o rio sem deixar que um grupo de missionários de um lado fique em número menor que o número de canibais nesse mesmo lado do rio.

a. Formule o problema fazendo apenas as especificações necessárias para assegurar uma solução válida. Faça um diagrama do espaço de estados completo. Pode ser interessante verificar a existência de estados repetidos.

Especificações do problema:

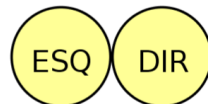
Vamos supor que começamos sempre na direita e queremos chegar na esquerda:

- Não podemos levar a mesma quantidade de missionários e canibais duas vezes consecutivas, para não entrarmos em um loop.
- Nas viagens da direita para esquerda, o ideal é levar o número máximo de tripulantes, já da esquerda para direita, o número mínimo.

- Não devemos começar levando apenas 1 missionário ou canibal, pois o segundo movimento seria repetido.
- Cada estado possui no máximo 5 sucessores, isso é definido de acordo com a capacidade do barco, que no caso é 2. Podemos levar, 1 canibal ou 1 missionário ou 2 canibais ou 2 missionários ou 1 canibal e 1 missionário.
- Para cada estado podemos gerar os 5 estados possíveis fazendo essas viagens especificadas acima, porém, é fato que isso ocasionará na criação de estados inválidos, portanto, antes de criar, verificamos se o estado é válido.
- Um estado será válido quando, existindo missionários na margem, o número de missionários for maior que o de canibais; a soma de missionários da esquerda e da direita deve ser igual a 3, análogo para os canibais. E o número de canibais e missionários, tanto na esquerda quanto na direita devem ser todos positivos, para eliminar casos como -4 e 1.
- Devemos guardar o estado anterior para tratar o caso das repetições e também os estados já visitados, para não visitarmos duas vezes.
- A cada novo estado visitado, verificamos se é um estado final, em caso, encerramos a busca, caso negativo, buscamos seus sucessores conforme indicado acima e inserimos estes em uma pilha de estados pendentes.
- Faremos isto iterativamente até que a lista de estados pendentes esteja vazia ou caso encontremos a solução.

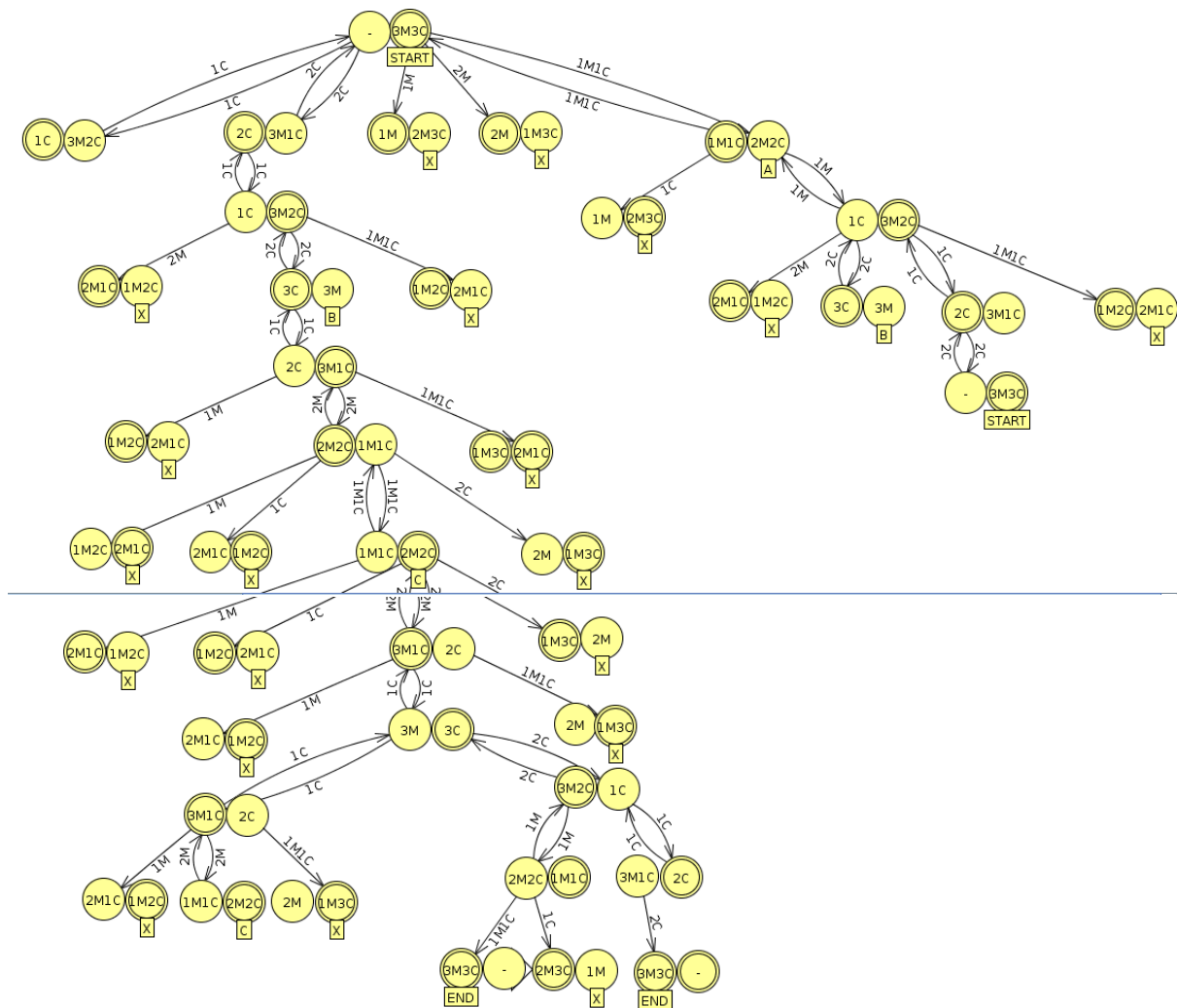
O diagrama de estados elaborado funciona da seguinte maneira:

Cada estado é composto por dois círculos, onde cada um deles representa uma das margens, esquerda e direita.



Já o barco é representado por um círculo circunscrito na margem em que ele estiver:





Explicação dos labels:

[START] Estado inicial.

[X] são referentes a estados inválidos que não geram sucessores.

[<letra>] é referente a estados repetidos, que possuem várias formas de se chegar neles.

[END] se refere ao estado final.