

Respostas dos Exercícios – Cap. 2 – Russell & Norvig

1. **(2.2)** Uma medida de desempenho é utilizada por um observador externo para avaliar o sucesso de um agente. Uma função de utilidade é utilizada por um agente para avaliar estados. A função de utilidade pode não ser igual a medida de desempenho; além disso, um agente pode não ter uma função de utilidade explícita, enquanto que sempre existe uma medida de desempenho.
2. **(2.3-a)** Sim. Podemos criar um novo programa de agente através da modificação de um programa de agente existente, inserindo-se comandos inúteis que não alterem a saída do programa. Estes dois programas implementam a mesma função de agente.
3. **(2.4-b)** O agente deve ter memória para guardar a informação de que um quadrado já foi limpo. Dessa forma ele só vai para o outro quadrado se ele ainda não verificou que o quadrado estava limpo. A memória que guarda essa informação é um estado interno.

4. **(2.5)**

Agente	Medida de Desempenho	Ambiente	Atuadores	Sensores
Robô jogador de futebol	Ganhar jogo, saldo de gols	Campo, bola, trave, próprio time, outro time, próprio corpo	Dispositivos para locomoção e chute	Câmera, sensores de toque, acelerômetros, sensores de orientação
Agente de compras na Internet	Obter produtos requisitados, minimizar gastos	Internet	Seguir link, preencher dados em campos, mostrar para usuário	Páginas da Web, pedidos dos usuários
Robô explorador de Marte	Terrenos explorados, amostras recolhidas	Veículo lançador, Marte	Rodas, equipamento coletor de amostras, transmissor de rádio	Câmera, sensores de toque, acelerômetros, sensores de orientação, receptor de rádio

5. **(2.6)**

Ambiente	Observável	Determinístico	Episódico	Estático	Discreto	Agente
Futebol	Parcialmente	Estocástico	Seqüencial	Dinâmico	Contínuo	Múltiplo
Internet	Parcialmente	Determinístico	Seqüencial	Estático	Discreto	Único
Marte	Parcialmente	Estocástico	Seqüencial	Dinâmico	Contínuo	Único
Teoremas	Totalmente	Determinístico	Seqüencial	Semi	Discreto	Múltiplo

Respostas dos Exercícios – Cap. 3 – Russell & Norvig

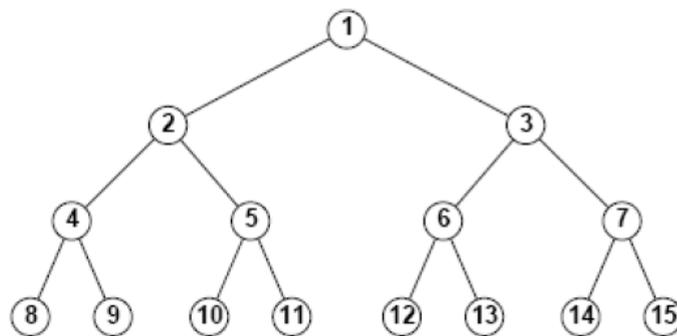
- (3.3)** $SUCCESSOR(s) = \{(a,s') \mid a \in ACOES-VALIDAS(s) \wedge s' = RESULTADO(a,s)\}$
 $ACOES-VALIDAS(s) = \{a \mid (a,s') \in SUCCESSOR(s)\}$
 $RESULTADO(a,s) = \{s' \mid (a,s') \in SUCCESSOR(s)\}$
- (3.6)** Não. Um espaço de estados finitos nem sempre leva a uma árvore de busca finita. Considere um espaço de estados com dois estados, cada um deles com uma ação que leva ao outro. Isso gera uma árvore de busca infinita, porque podemos ir e voltar infinitas vezes. Porém, se o espaço de estados for uma árvore finita, ou, em geral, um DAG (grafo acíclico direcionado), não haverá loops e a árvore de busca será finita.

3. **(3.7)**

- Estado inicial: nenhuma região colorida.
Teste de objetivo: todas as regiões coloridas, e nenhuma região adjacente com a mesma cor.
Função sucessor: atribuir uma cor a uma região que esteja sem cor.
Função de custo: número total de atribuições (e igual para todas as soluções).
- Estado inicial: como descrito no enunciado.
Teste de objetivo: macaco alcançou as bananas.
Função sucessor: subir no engradado; descer do engradado; mudar engradado de lugar; andar de um lugar a outro; agarrar bananas
Função de custo: número total de ações.

4. **(3.8)**

a.

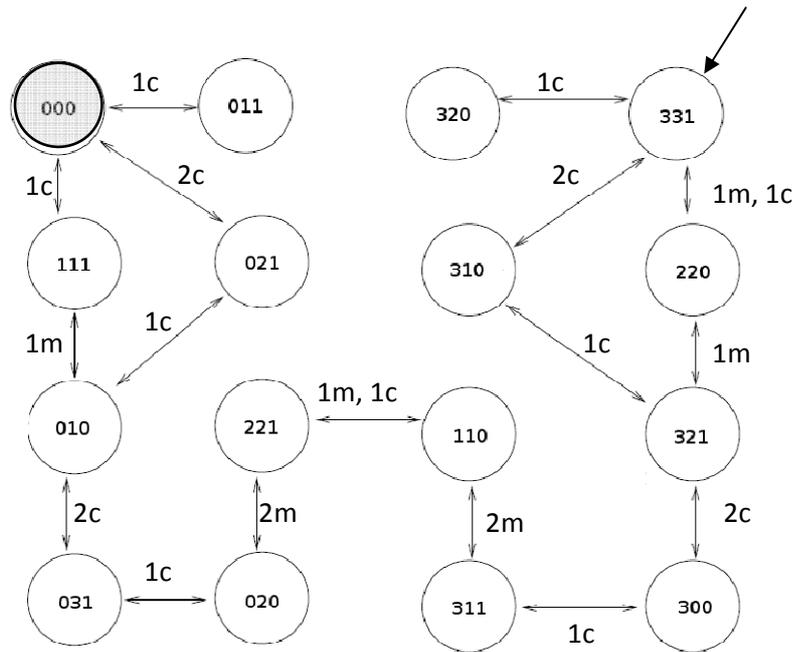


- Busca em extensão: 1 2 3 4 5 6 7 8 9 10 11
Busca em profundidade limitada: 1 2 4 8 9 5 10 11
Aprofundamento iterativo: 1; 1 2 3; 1 2 4 5 3 6 7; 1 2 4 8 9 5 10 11

5. (3.9)

- a. Uma representação possível: Um estado é um vetor com três inteiros listando o número de missionários, canibais e barcos na margem inicial do rio. Sendo assim, o estado inicial é (3,3,1) e o estado objetivo é (0,0,0). O teste de objetivo verifica se o estado objetivo (0,0,0) foi alcançado. A função de custo tem valor um para cada ação. Os sucessores de um estado são todos os estados que movem uma ou duas pessoas e um barco de um lado para o outro, sem que o número de missionários de um lado fique menor do que o número de canibais.

Espaço de Estados:



- b. Qualquer algoritmo de busca funciona bem, porque o espaço de estados é muito pequeno. Basta eliminar estados repetidos e estados inválidos (com maior número de canibais do que missionários do mesmo lado).

Respostas dos Exercícios – Cap. 4 – Russell & Norvig

1. (4.1) Sequência de nós na borda:

L[0+244=244]
M[70+241=311], T[111+329=440]
L[140+244=384], D[145+242=387], T[111+329=440]
D[145+242=387], T[111+329=440], M[210+241=451], T[251+329=580]
C[265+160=425], T[111+329=440], M[210+241=451], M[220+241=461], T[251+329=580]
T[111+329=440], M[210+241=451], M[220+241=461], P[403+100=503], T[251+329=580], R[411+193=604],
D[385+242=627]
M[210+241=451], M[220+241=461], L[222+244=466], P[403+100=503], T[251+329=580], A[229+366=595],
R[411+193=604], D[385+242=627]
M[220+241=461], L[222+244=466], P[403+100=503], L[280+244=524], D[285+242=527], T[251+329=580],
A[229+366=595], R[411+193=604], D[385+242=627]
L[222+244=466], P[403+100=503], L[280+244=524], D[285+242=527], L[290+244=534], D[295+242=537],
T[251+329=580], A[229+366=595], R[411+193=604], D[385+242=627]
P[403+100=503], L[280+244=524], D[285+242=527], M[292+241=533], L[290+244=534], D[295+242=537],
T[251+329=580], A[229+366=595], R[411+193=604], D[385+242=627], T[333+329=662]
B[504+0=504], L[280+244=524], D[285+242=527], M[292+241=533], L[290+244=534], D[295+242=537], T[251+329=580],
A[229+366=595], R[411+193=604], D[385+242=627], T[333+329=662], R[500+193=693], C[541+160=701]

2. (4.2) $w=0$ faz com que $f(n) = 2g(n)$, que equivale à busca de custo uniforme (a multiplicação por 2 não modifica a ordem em que os nós são expandidos). $w=1$ faz com que $f(n)=g(n)+h(n)$, que equivale à busca A^* . $w=2$ faz com que $f(n) = 2h(n)$, que equivale à busca gulosa pela melhor escolha. Este algoritmo é ótimo quando $h(n)$ é admissível e $w \leq 1$.

3. (4.3)

- Quando todos os custos são iguais, temos que $g(n)$ é proporcional a $\text{profundidade}(n)$, logo a busca de custo uniforme reproduz a busca em extensão, já que os nós são expandidos em ordem de menor profundidade (menor custo) para maior profundidade.
- Busca em extensão equivale à busca pela melhor escolha com $f(n)=\text{profundidade}(n)$; busca em profundidade e busca pela melhor escolha com $f(n)=-\text{profundidade}(n)$; busca de custo uniforme e busca pela melhor escolha com $f(n)=g(n)$.
- Busca de custo uniforme equivale a A^* com $h(n)=0$.

4. (4.9) A heurística dos blocos mal posicionados é exata para o problema em que um bloco pode ser movido de um quadrado A para qualquer quadrado B. Como esse problema é um relaxamento da condição que um quadrado pode ser movido do quadrado A para um quadrado B se B estiver vazio, o valor da heurística de Gaschnig não pode ser menor que o valor da heurística dos blocos mal-posicionados. Como ela também é admissível (por ser um relaxamento do problema original), ela é mais precisa.

Se permutarmos dois blocos adjacentes no estado objetivo, teremos um estado em que a heurística dos blocos mal-posicionados e a heurística da distancia Manhattan terão valor 2, e a heurística de Gaschnig terá valor 3.

Para calcular a heurística de Gaschnig, repita o seguinte ate que o estado objetivo seja atingido: seja B a posição atual do espaço vazio; se B for ocupado pelo quadrado X (não vazio) no estado objetivo, mova X para B; senão, mova qualquer bloco mal posicionado para B.

5. **(4.11)**

- a. Busca de subida de encosta.
- b. (Não faz sentido)
- c. Busca de subida de encosta.
- d. Busca aleatória.