Nome: Luís Felipe de Melo Costa Silva
Número USP: 9297961

Lista de Exercícios 1 - MAC0425

Exercício 3.3

a) Para formular um problema de busca precisamos dos seguintes itens:

- Estados do mundo: Cidades do mapa.
- Ações: Ir de uma cidade para outra.
- Função de transição de estado: Dado um dos amigos e a cidade onde ele está, junto com uma cidade adjacente, devolve esse amigo na cidade adjacente com custo igual a $d(i, j)$.
- Função de custo de caminho: Soma de vários termos na forma $max(d(i, j), d(k, l))$, sendo i e k a cidade onde A e B estão, respectivamente, em uma iteração e j e l, cidades adjacentes a i e k respectivamente, não necessariamente diferentes. Os termos tem essa forma porque um amigo espera o outro terminar o caminho antes do próximo turno começar.
- Um estado inicial: A e B nas cidades em que vivem.
- Estados meta: As cidades em que A e B podem se encontrar.
- Teste de meta: Se a cidade em que A está for a mesma que B está, essa cidade é um estado meta.

b) As únicas heurísticas admissíveis são (i)$D(i, j)$ e (iii)$\frac{D(i,j)}{2}$. Uma heurística admissível é aquela que nunca ultrapassa o custo real de se alcançar a meta. Portanto, usando heurísticas menores ou iguais a uma linha reta (que é a menor distância entre dois pontos), teremos heurísticas admissíveis.

Exercício 3.11

Um **estado do mundo** é uma situação concreta no mundo real. Já uma **descrição do estado** é uma representação abstrata do mundo real que é usadas por agentes de busca. Por último, um **nó de busca** é um nó numa árvore de busca, onde a raiz é o estado inicial e os filhos de cada nó são os estados alcançáveis a partir de ações. Essa distinção é útil na modelagem dos problemas de busca. Utilizamos os estados do mundo para entender o problema e então, criamos as descrições dos estados. Com isso, procuramos a melhor abordagem para resolver o problema. Na implementação de um algoritmo para solucionar o problema, geralmente trabalhamos com grafos, por isso é útil usarmos nós de busca.

Exercício 3.29

Sabemos que uma heurística h é consistente se sua estimativa é sempre menor ou igual à distância estimada de qualquer nó vizinho até o objetivo mais o custo de chegar nesse vizinho, ou seja:

$$h(n) \leq c(n, v) + h(v),$$

onde v é um nó sucessor de n e $c(x)$ é a função de custo.

Uma heurística admissível h é aquela que nunca ultrapassa o real custo h^* de se alcançar a meta, em outros termos:

$$h(n) \leq h^*(n)$$

Vamos provar por indução que uma heurística consistente também é admissível.

Base: Seja u um nó antecessor de v, que é o estado meta nesse caso. Como a heurística é consistente:

$$h(u) \leq c(u, v) + h(v) = c(u, v) + 0 = c(u, v)$$

Passo: Seja t um nó. O custo ótimo para alcançar v de t é $h^*(t)$. Ele é calculado como $min_{x \in A}(c(t, x) + h^*(x))$, onde A é o conjunto de sucessores de t. Como a heurística é consistente, então $h(t) \leq c(t, t') + h(t')$. Além disso,

como $h(t) \leq h^*(t)$ é o que estamos queremos provar, $h(t) \leq c(t, t') + h^*(t)$, e isso é verdade para todos os sucessores t' do nó t. Em outras palavras, $h(t) \leq min_{x \in A}(c(t, x) + h^*(x)) = h^*(t)$, logo, $h(t) \leq h^*(t)$. □

Exercício Extra 1

a) Usando a BFS (busca em largura), a ordem dos nós a serem visitados será definida por uma fila (o primeiro a entrar é o primeiro a sair). Paramos de expandir os nós quando o estado meta está na borda (sétima linha). Tabela 1.

b) Usando a DFS (busca em profundidade), a ordem dos nós a serem visitados será definida por uma pilha (o último a entrar é o primeiro a sair). Paramos de expandir os nós quando o estado meta é visitado (sétima linha). Tabela 2.

Table 1: Expandindo com BFS

Nós expandidos	**Borda**
	A
A	B, D, G
A, B	D, G
A, B, D	G, C, E
A, B, D, G	C, E, K, L
A, B, C, D, G	E, K, L, F
A, B, C, D, E, G	K, L, F, I
A, B, C, D, E, G, K	L, F, I, O
A, B, C, D, E, G, K, L	F, I, O, H
A, B, C, D, E, F, G, K, L	I, O, H, J
A, B, C, D, E, F, G, I, K, L	O, H, J, M
A, B, C, D, E, F, G, I, K, L, O	H, J, M
A, B, C, D, E, F, G, H, I, K, L, O	J, M
A, B, C, D, E, F, G, H, I, J, K, L, O	M
A, B, C, D, E, F, G, H, I, J, K, L, M, O	N
A, B, C, D, E, F, G, H, I, J, K, L, M, N, O	

Table 2: Expandindo com DFS

Nós expandidos	**Borda**
	A
A	B, D, G
A, B	D, G
A, B, D	C, E, G
A, B, C, D	F, E, G
A, B, C, D, F	I, J, E, G
A, B, C, D, F, I	M, J, E, G
A, B, C, D, F, I, M	N, J, E, G
A, B, C, D, F, I, M, N	J, E, G
A, B, C, D, F, I, J, M, N	E, G
A, B, C, D, E, F, I, J, M, N	G
A, B, C, D, E, F, G, I, J, M, N	K, L
A, B, C, D, E, F, G, I, J, K, M, N	O, L
A, B, C, D, E, F, G, I, J, K, M, N, O	L
A, B, C, D, E, F, G, H, I, J, K, L, M, N, O	