Ministério da Educação
Universidade Federal do Agreste de Pernambuco

DISCIPLINA: Inteligência Artificial – Instância 2020.2
PROFESSOR: Ryan Ribeiro de Azevedo
ALUNO: Armstrong Lohãns de Melo Gomes Quintino
FINAL

2. a) Explique a Busca em Largura (0,5)

A busca em largura é um algoritmo de busca em grafos utilizado para realizar uma busca ou travessia num grafo e estrutura de dados do tipo árvore. Intuitivamente, você começa pelo vértice raiz e explora todos os vértices vizinhos. Então, para cada um desses vértices mais próximos, exploramos os seus vértices vizinhos inexplorados e assim por diante, até que ele encontre o alvo da busca.

Formalmente, uma busca em largura é um método de busca não-informada que expande e examina sistematicamente todos os vértices de um grafo direcionado ou não-direcionado. Em outras palavras, podemos dizer que o algoritmo realiza uma busca exaustiva num grafo passando por todas as arestas e vértices do grafo.

Sendo assim, o algoritmo deve garantir que nenhum vértice ou aresta seja visitado mais de uma vez e, para isso, utiliza uma estrutura de dados fila para garantir a ordem de chegada dos vértices. Dessa maneira, as visitas aos vértices são realizadas através da ordem de chegada na estrutura fila e um vértice que já foi marcado não pode entrar novamente a esta estrutura.

3. a) Explique a Busca A* (0,5)

Algoritmo A* é um algoritmo para Busca de Caminho. Ele busca o caminho em um grafo de um vértice inicial até um vértice final. Ele é a combinação de aproximações heurísticas como do algoritmo de Busca em Largura e da formalidade do Algoritmo de Dijkstra.

O algoritmo plota com eficiência um caminho percorrível entre vários nós, ou pontos, no gráfico. Em um mapa com muitos obstáculos, encontrar caminhos dos pontos **A** a **B** pode ser difícil.

No entanto, o algoritmo A* introduz uma heurística em um algoritmo de busca de gráfico regular, essencialmente planejando com antecedência em cada etapa para que uma decisão mais otimizada seja feita.

Como Dijkstra, A* funciona criando uma árvore de caminho de custo mais baixo do nó inicial ao nó de destino. O que torna A* diferente e melhor para muitas pesquisas é que, para cada nó, A* usa uma função **f(n)** que fornece uma estimativa do custo total de um caminho usando aquele nó. Portanto, A* é uma função heurística, que difere de um algoritmo porque uma heurística é mais uma estimativa e não é necessariamente comprovadamente correta.

Sua aplicação vai desde aplicativos para encontrar rotas de deslocamento entre localidades a resolução de problemas, como a resolução de um quebra-cabeças. Ele é muito usado em jogos.

4. Formalizações:
a) ∀p CompraPorAlquere(p, cenoura) → Possui(p, coelho) ∨ Possui(p, mercearia)
b) ∀b Cão(b) → Persegue(b, coelho)
c) CompraPorAlquere(Maria, cenoura)
d) ∀p Possui(p, coelho) → ∀b ∀c Coelho(b) → Persegue(c, b) → Odeia(p, c)
e) Possui(João, cão)
f) ∀p ∀a ∀c Odeia(a, Possui(p, c)) → ¬ Namora(a, p)
g) ¬ Possui(Maria, mercearia) -> ¬ Namora(Maria, João)

(Não deu tempo de finalizar o Tableaux)