

**Identificação do aluno:**

Nome: \_\_\_\_\_ #: \_\_\_\_\_

**I**

1. Analise as seguintes funções escritas em Python e explique o que fazem:

a) 

```
def f(x):  
    if x==[]:  
        return 0  
    if x[0]>0:  
        return x[0] + f(x[1:])  
    return f(x[1:])
```

b) 

```
def g(x):  
    if x==[]:  
        return [[]]  
    y = g(x[1:])  
    return y + [ [x[0]]+z for z in y ]
```

c) 

```
def h(x,y):  
    return [ z for z in x if not y(z) ]==[]
```

2. Implemente em Python as seguintes funcionalidades relevantes para sistemas inteligentes:

a) No contexto da geração de todas as interpretações de uma fórmula em lógica proposicional, é necessário gerar todas as combinações de valores possíveis das diversas variáveis proposicionais contidas na fórmula. Assim, programe uma função que, dada uma lista de variáveis proposicionais, gere todas as combinações de valores possíveis.

Exemplo:

```
>>> interpretacoes(["a","b"])  
[ [ ("a",True), ("b",True) ], [ ("a",True), ("b",False) ], [ ("a",False), ("b",True) ], [ ("a",False), ("b",False) ] ]
```

b) Considere que se representa o conhecimento relevante para uma aplicação através de um conjunto de fórmulas em lógica proposicional, podendo cada fórmula ser uma proposição (facto) ou uma implicação (regra) na forma seguinte:  $A_1 \wedge \dots \wedge A_n \Rightarrow C$ , em que  $A_1, \dots, A_n$  e  $C$  são proposições. Para efeitos de implementação em Python, uma fórmula é representada na forma de uma lista  $[C, A_1, \dots, A_n]$ . Uma lista com apenas um elemento, representa um facto. Programe uma função que, dado conhecimento na forma de uma lista de fórmulas, e dada uma proposição, retorne o valor lógico da proposição. Uma proposição que não pode ser provada como verdadeira é considerada falsa.

Exemplo:

```
>>> provar([["a"], ["b"], ["c", "a", "b"]], "c")
```

```
True
```

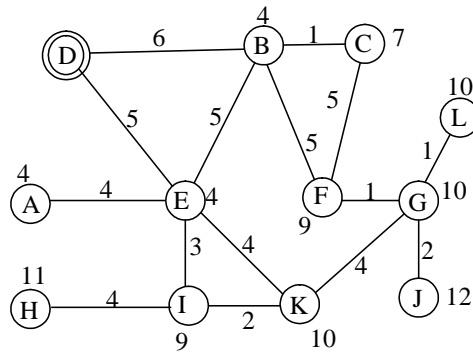
c) No contexto da implementação de algoritmos de pesquisa em árvore, considere que os nós da árvore são representados por tuplos  $(Id, State, IdParent)$ , em que  $Id$  é um identificador do nó,  $State$  é o estado desse nó e  $IdParent$  é o identificador do nó pai. Considerando que se está a usar uma estratégia de pesquisa em largura, programe uma função que, dada a lista de nós abertos (nós que aguardam expansão) e dada uma lista de novos nós (que acabam de ser criados por expansão de outro nó), produza a nova lista de nós abertos.

II

1. Identifique semelhanças e diferenças entre a pesquisa em árvore em profundidade e a pesquisa por montanhismo.

2. As casas têm divisões de diferentes tipos, por exemplo, salas de estar, salas de jantar, quartos de dormir, cozinhas e quartos de banho. As divisões da casa têm peças de mobiliário, como por exemplo, mesas, cadeiras, camas, cómodas e estantes. A casa da Gabriela é em Aveiro. Essa casa tem um quarto com uma cama, em que a Gabriela dorme, e uma cómoda. Represente este conhecimento através de uma rede semântica.

3. O grafo a seguir apresentado representa um espaço de estados num problema de pesquisa, sendo **D** o estado objectivo (solução). As estimativas do custo de chegar à solução a partir de cada estado estão anotadas junto aos mesmos. Os custos das transições estão anotados junto às ligações do grafo.



a) Verifique se as estimativas de custo anotadas junto a cada nó constituem uma heurística admissível para a pesquisa  $A^*$ . Se não for esse o caso, introduza (na própria figura) alterações que a tornem admissível. Justifique.

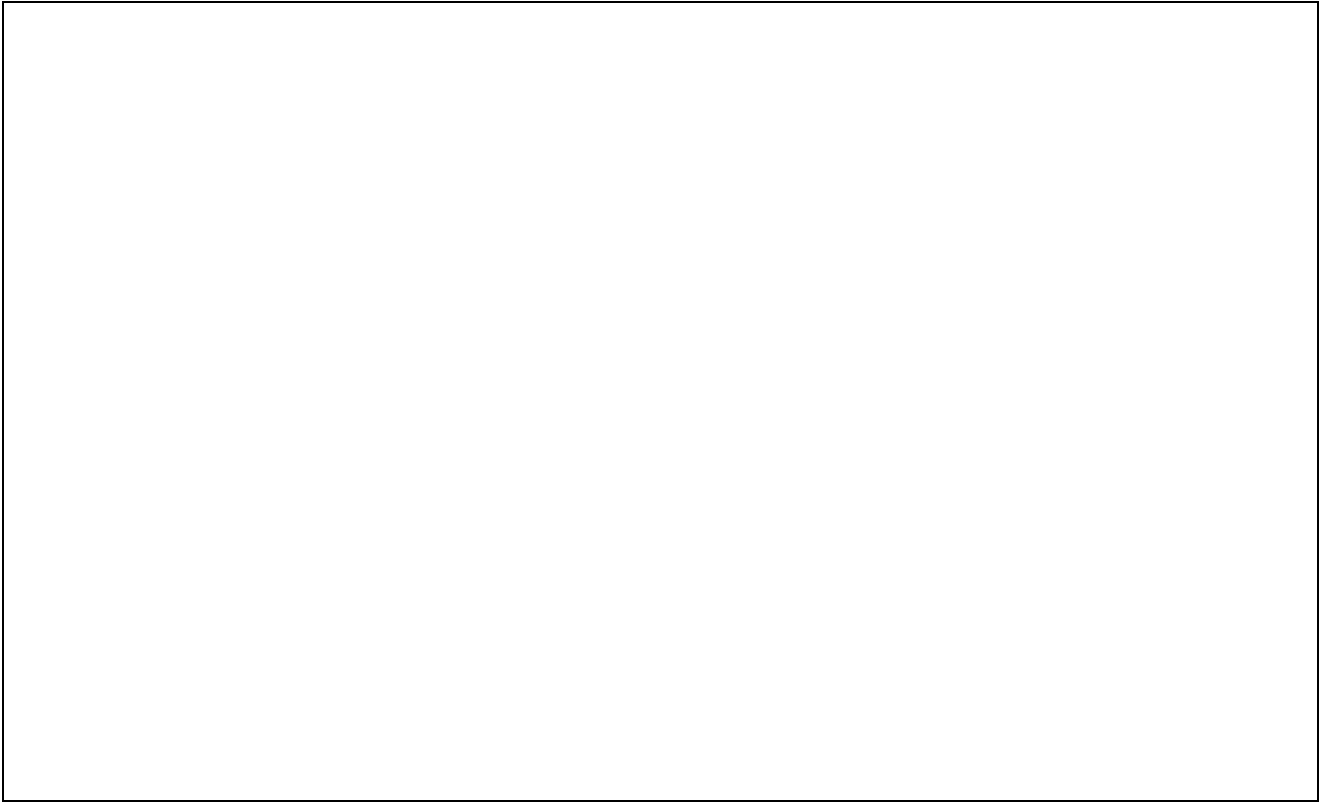
b) Tomando o estado **G** como estado inicial, apresente a árvore de pesquisa gerada quando se realiza uma pesquisa  $A^*$  com repetição de estados. Numere os nós pela ordem em que são acrescentados à árvore e anote também o valor da função de avaliação em cada nó. Em caso de empate nos valores da função de avaliação em dois ou mais nós, utilize a ordem alfabética dos respectivos estados.

c) Indique um valor aproximado do factor de ramificação efectivo da árvore gerada. Justifique.

4. Considere um veículo autónomo que se movimenta num ambiente estruturado em nós e ligações, ou seja, estruturado como um grafo. As ligações correspondem a ruas. Os nós representam confluências de uma ou mais ruas. Além disso, em cada nó pode haver 0 ou mais parques de estacionamento. As ruas começam e terminam em nós adjacentes no grafo. O agente é capaz de realizar as seguintes acções: atravessar (passar para outra rua do nó), estacionar num dos parques do mesmo nó, percorrer (seguir até ao nó no outro extremo da rua actual), sair do estacionamento para uma dada rua que começa ou termina no mesmo nó.

a) Identifique e caracterize um conjunto de predicados em lógica de primeira ordem que possam ser usados para especificar condições sobre estados de planeamento neste domínio. Identifique os valores possíveis dos argumentos desses predicados. ( Nota: Para responder a esta pergunta, é aconselhável ver também a alínea b), onde estes predicados também são usados. )

b) Usando os predicados que propôs, defina um conjunto de operadores STRIPS para representar as acções que podem ser realizadas neste domínio.



***Identificação do aluno:***

Nome: \_\_\_\_\_ #: \_\_\_\_\_

#INTRODUÇÃO À INTELIGÊNCIA ARTIFICIAL

Exame, xx/xx/xxxx (Tempo: 3h)