

**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:])
```

A função  $f$  realiza uma soma acumulativa dos elementos positivos de uma lista. Se a lista estiver vazia, retorna 0. Se o primeiro elemento for positivo, adiciona esse elemento à soma e continua a recursão com os elementos restantes da lista. Se o primeiro elemento não for positivo, simplesmente continua a recursão com os elementos restantes da lista. Essa função, portanto, calcula a soma dos elementos positivos de uma lista.

b)

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

A função  $g(x)$  gera todas as combinações possíveis dos elementos na lista  $x$ . Essa é uma implementação recursiva que utiliza a técnica de divisão e conquista para gerar todas as combinações.

2. 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) ] ]
```

```
def interpretacoes(variaveis):  
    if not variaveis:  
        return []  
  
    resto_combinacoes = interpretacoes(variaveis[1:])  
  
    return [(variaveis[0], True)] + combinacao for combinacao in resto_combinacoes +  
    \[(variaveis[0], False)] + combinacao for combinacao in resto_combinacoes]
```

--	--

## II

1. Neste exercício, tem um conjunto de questões de escolha. Em cada alínea, apenas uma das opções dadas está certa, e apenas pode seleccionar uma delas. Cada resposta errada desconta 20% da cotação da alínea.

a) A frase “Todos os livros de Banda Desenhada têm capa dura” pode ser representada em Lógica de 1ª ordem da seguinte forma:

- $\forall x (\text{Livro}(x) \wedge \text{BandaDesenhada}(x)) \Rightarrow \neg \text{CapaDura}(x)$
- $\forall x \text{ BandaDesenhada}(x) \Rightarrow \neg \text{Capa}(x, \text{Dura})$
- $\forall x \text{ Livro}(x) \vee (\text{BandaDesenhada}(x)) \Rightarrow \neg \text{Capa}(x, \text{Dura})$
- $\forall x \text{ Livro}(x) \wedge (\text{BandaDesenhada}(x)) \Rightarrow \neg \text{Capa}(x, \text{Dura})$
- Nenhuma das anteriores

X

b) A frase “A melhor nota a Português foi a da Ana” pode ser representada em Lógica de 1ª ordem da seguinte forma:

- $\forall x \text{ Nota}(\text{Ana}, \text{Português}) > \text{Nota}(x, \text{Português}) \wedge \text{Aluno}(x)$
- $\forall x, y, z \text{ Nota}(\text{Ana}, \text{Português}, y) > \text{Nota}(x, \text{Português}, z) \wedge y > z$
- $\forall x \text{ Nota}(\text{Ana}, \text{Português}) \geq \text{Nota}(x, \text{Português}) \vee \text{Aluno}(x)$
- $\forall x \text{ Aluno}(x) \Rightarrow (\text{Nota}(\text{Ana}, \text{Português}) \geq \text{Nota}(x, \text{Português}))$
- Nenhuma das anteriores

X

c) Pesquisa por melhorias sucessivas é:

- Uma técnica de pesquisa para resolução problemas de atribuição
- Uma técnica para combinação de heurísticas
- Uma técnica de pesquisa para optimização de soluções
- Um caso particular de recozimento simulado em que a evolução da temperatura faz lembrar uma paisagem de montanhas
- Nenhuma das anteriores

X

d) Uma consequência lógica do conjunto de fórmulas  $\{ A \vee B, \neg B \vee C \vee D, \neg A, \neg D \}$  é:

- $B \wedge A$
- C
- A
- $A \vee D$
- Nenhuma das anteriores

X

e) Os operadores STRIPS são:

- Um formato de representação de acções para sistemas reactivos com estado interno
- Um formato de representação de transições de estados para pesquisa por melhorias sucessivas
- Mecanismos de modificação da solução em pesquisa por recozimento simulado
- Mecanismos para geração de planos no mundo dos blocos
- Nenhuma das anteriores

X

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

**\*\*Semelhanças:\*\***

Depth-First Search(DFS)=Busca em Profundidade

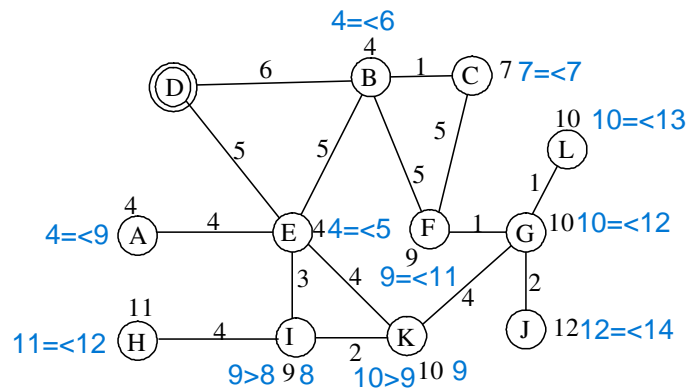
- Ambos exploram espaços de estados em busca de soluções.
- Não garantem solução ótima.

**\*\*Diferenças:\*\***

- DFS segue "profundidade primeiro", enquanto a busca por montanhismo usa heurística para escolher o caminho mais promissor.
- DFS usa backtracking; busca por montanhismo se move em direção ao vizinho mais promissor.
- DFS é completo; busca por montanhismo pode ficar presa em ótimos locais locais.
- DFS explora todo o caminho em uma ramificação; busca por montanhismo foca-se no objetivo, ignorando partes não relevantes.
- DFS não precisa armazenar explicitamente estados visitados; busca por montanhismo frequentemente requer lista de estados visitados.
- DFS é usado em problemas de busca; busca por montanhismo é aplicada em problemas de otimização.

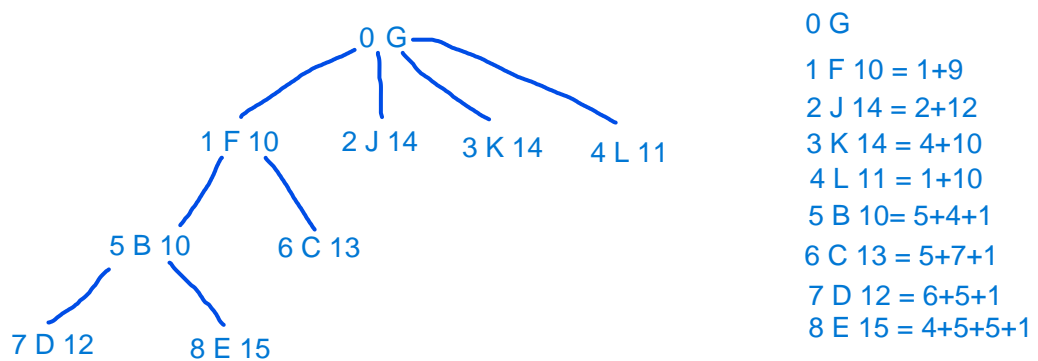
3. As casas têm divisões de diferentes tipos, por exemplo, salas de estar, salas de jantar, quartos de dormir, cozinhas e quartos de banho. As divisões da casa têm peças de mobiliário, como por exemplo, mesas, cadeiras, camas, cómodas e estantes. A casa da Gabriela é em Aveiro. Essa casa tem um quarto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

4. 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.

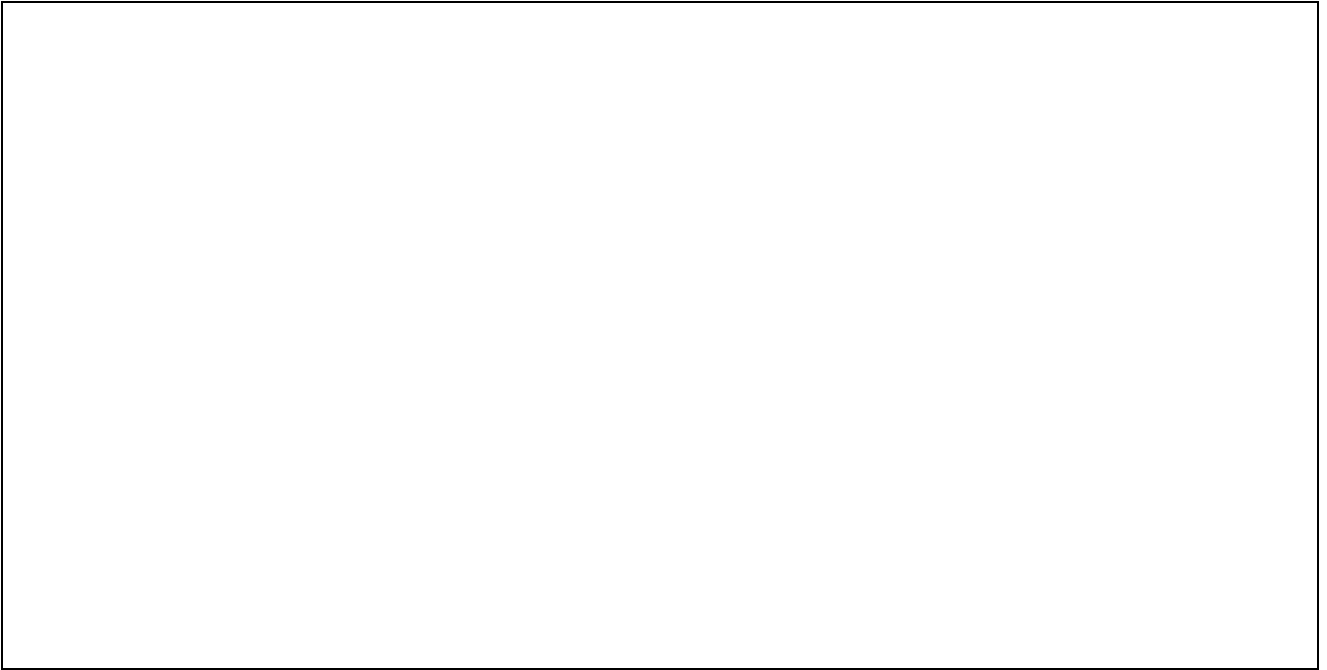
b) Tomando o estado **G** como estado inicial, apresente a árvore de pesquisa gerada quando se realiza uma pesquisa A\* com repetição de estados. Numere os nós pela ordem em que são acrescentados à árvore e anote também o valor da função de avaliação em cada nó. Em caso de empate nos valores da função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



5. 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.)

atravessar(veiculo, nó, rua)  
estacionar(veiculo, nó, parque)  
percorrer(veiculo, rua)  
sairestacionamento(veiculo, rua, nó)



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

# INTELIGÊNCIA ARTIFICIAL

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