

# I

1. Analise as seguintes funções escritas em Python e explique o que fazem. Não precisa descrever o funcionamento interno das funções.

a) 

```
def f(x,y):  
    return [z for z in x if y(z)]!=[]
```

Dado uma função <sup>recursiva</sup> ~~f~~ que contém como parâmetros ~~uma~~ lista ~~(x,y)~~ e ~~n~~ e uma função ~~y~~ e é retornado true se ~~n~~ tiver um elemento que verifique a condição ~~y~~ (não ser vazio)

b) 

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

Dado uma função ~~g~~ recursiva que contém como parâmetros duas listas ~~(n,y)~~  
~~Assim~~ Assim como resultado desta função é devolver o número de vezes que um dado elemento ~~n~~ aparece na lista ~~y~~, devolvendo também ~~a~~ a lista ~~y~~ atualizada, ou seja, sem os elementos que são iguais a ~~n~~.  
Devolverá então um tuple em que o primeiro elemento é o número de vezes que ~~n~~ aparece na lista ~~y~~ e o segundo elemento é a lista ~~y~~ sem os elementos que são iguais a ~~n~~.

2. Implemente em Python o algoritmo de pesquisa por montanhismo. Para esse efeito, deverá programar uma função que recebe como entrada: uma solução inicial; uma função que gera uma lista de novas soluções obtidas por modificação de uma dada solução; e uma função que avalia uma dada solução. Será valorizada a utilização de funções de ordem superior pré-definidas.

```
def hill_climbing(initial_solution, generate_neighbors, evaluate):  
    current = initial_solution  
    while True:  
        neighbors = generate_neighbors(current)  
        best_neighbor = max(neighbors, key=evaluate)  
        if evaluate(best_neighbor) <= evaluate(current):  
            return current  
        current = best_neighbor
```

```
initial_solution = [1, 2, 3]
```

```
def generate_neighbors(solution):  
    neighbors = []  
    for i in range(len(solution)):  
        neighbor = solution.copy()  
        neighbor[i] += random.choice([-1, 1])  
        neighbors.append(neighbor)  
    return neighbors
```

```
def evaluate(solution):  
    return sum(solution)
```

```
hill_climbing(initial_solution, generate_neighbors, evaluate)
```



## 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 "O pai do António é tio do Diogo." pode ser representada em Lógica de Primeira Ordem da seguinte forma:

- $\forall x \text{ Pai}(x, \text{Antonio}) \wedge \text{Tio}(x, \text{Diogo})$
- $\forall x \text{ Pai}(x, \text{Antonio}) \Rightarrow \text{Tio}(x, \text{Diogo})$
- $\exists x \text{ Pai}(x, \text{Antonio}) \wedge \forall y \text{ Nome}(y, \text{Diogo}) \Rightarrow \text{Tio}(x, y)$
- $\exists x \text{ Pai}(x, \text{Antonio}) \wedge \text{Tio}(x, \text{Diogo})$
- Nenhuma das anteriores

X

✓ b) Relativamente às redes de Bayes, indique a afirmação verdadeira

- Não permitem representar conhecimento impreciso
- São representadas por grafos dirigidos
- Permitem representar as dependências entre as variáveis através de predicados
- Permitem representar relações de herança entre entidades
- Nenhuma das anteriores

X

✓ c) Relativamente às redes semânticas, indique a afirmação verdadeira

- Permitem representar hierarquias de tipos
- Não permitem representar conhecimento por omissão
- Nenhum tipo de rede semântica permite representar a negação ou a disjunção
- A rede semântica estudada na componente prática da disciplina é tão expressiva como a lógica de primeira ordem
- 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:

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

X

✓ e) Uma diferença entre a pesquisa em largura e a pesquisa de custo uniforme é que:

- A pesquisa de custo uniforme expande apenas o ramo da árvore mais promissor
- A pesquisa de custo uniforme trabalha com um grafo de restrições
- A pesquisa em largura expande sempre o nó com menor custo
- A pesquisa em largura gere a lista nós abertos segundo a disciplina FIFO
- Nenhuma das anteriores

X

2. Considere o futebol multi-bola, variante do futebol em que existem múltiplas bolas em jogo ao mesmo tempo. Quando uma bola sai do campo ou entra numa baliza, um robô apanha-bolas leva-a para um depósito situado junto ao meio campo. Quando este robô não tem nenhuma bola, procura uma bola fora do campo para apanhar. Quando encontra uma bola fora do campo, agarra-a. De seguida, desloca-se para o depósito. Quando chega ao



depósito, coloca a bola no depósito, e volta a procurar bolas. No entanto, quando detecta que o depósito está cheio, aguarda que deixe de estar cheio, e só depois coloca lá a bola.

a) Identifique e caracterize as várias condições ou predicados que podem ser usadas para descrever as situações em que se pode encontrar o robô apanha-bolas. Identifique também quaisquer variáveis de estado que sejam necessárias.

~~RoboEm(x)~~  $x \in [foco]$   
 $RoboEm(x)$ ,  $x \in [foco campo, bolazo]$   
 $TorBola$   
 $BolaEm(x)$ ,  $x \in [deposito]$   
 $DepositoCheio$

b) Especifique um conjunto de regras situação-acção que definam um comportamento adequado do robô apanha-bolas. Pode fazê-lo na forma de uma tabela com as seguintes colunas:

- Situação - uma conjunção de condições, das que identificou na alínea a)
- Actualização - actualização das variáveis de estado, das que identificou na alínea a), se alguma
- Ação - acção a executar pelo agente na situação indicada

Situação	Actualização	Ação
$BolaEm(foco campo) \vee BolaEm(bolazo)$	—	Apanhar Bola
$TorBola$	—	Desloca-se
$\neg TorBola$	—	Procurar Bola
$BolaEm(deposito) \wedge \neg DepositoCheio$	—	<del>Depositar Bola</del> Aguardar
<del>DepositoCheio</del> $BolaEm(deposito) \wedge DepositoCheio$	—	<del>Aguardar</del> Aguardar deposição Bola
<del>DepositoCheio</del>		

c) Identifique e caracterize os principais tipos de agentes inteligentes bem como pelo menos uma arquitectura hierárquica para agentes mais complexos.

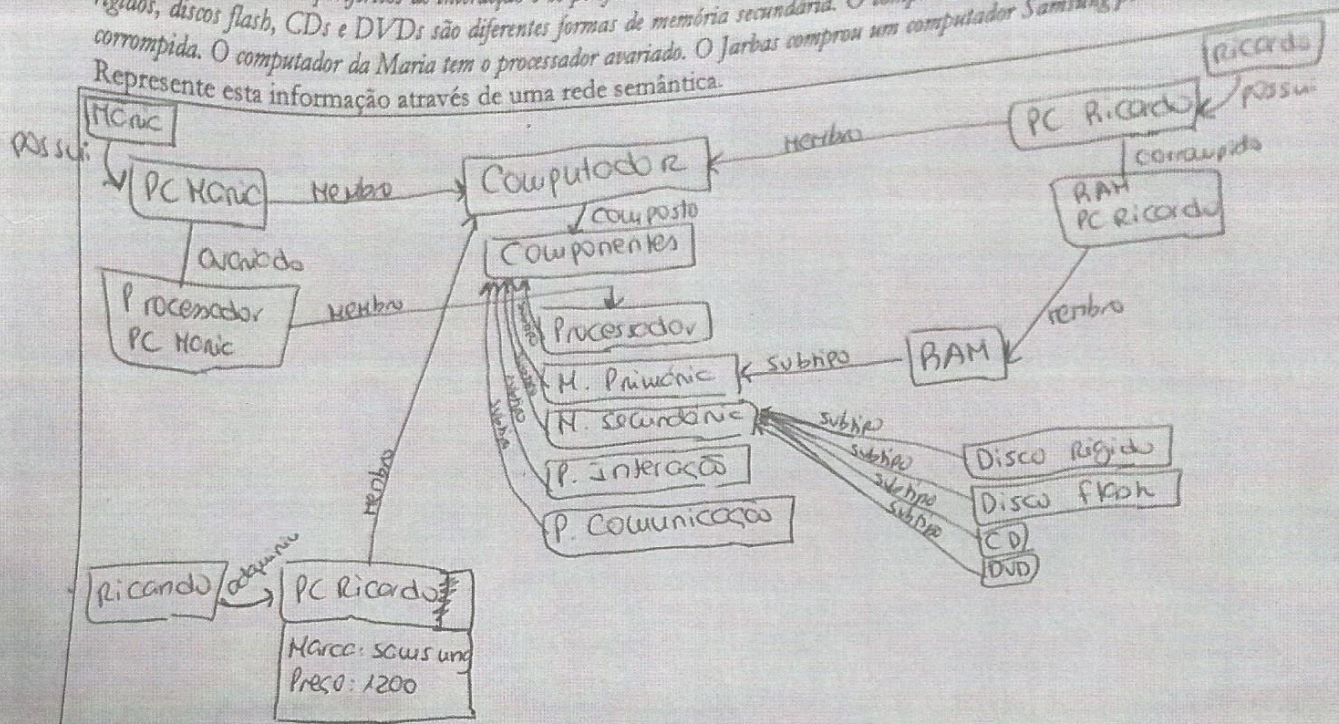
Os tipos de agentes inteligentes são: reactivo simples, reactivo com estado, deliberativo orientado para objecto e deliberativo orientado por funções de utilidade.  
 As arquitecturas são: subjugadas, três níveis, três camadas e CAEL.



3.

a) Considere a seguinte informação: "Os componentes principais de um computador são o processador, a memória primária, a memória secundária, os periféricos de interação e os periféricos de comunicação. A RAM é um tipo de memória primária. Os discos rígidos, discos flash, CDs e DVDs são diferentes formas de memória secundária. O computador do Ricardo tem a memória RAM corrompida. O computador da Maria tem o processador avariado. O Jarbas comprou um computador Samsung por 1200 euros."

Represente esta informação através de uma rede semântica.

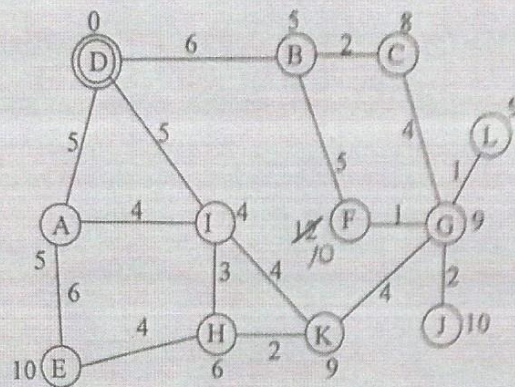


b) Explica o que se entende por Engenharia do Conhecimento.

Uma base de conhecimento é o conjunto de representações de fatores e regras de funcionamento do mundo.  
Engenharia do conhecimento é o processo ou atividade de construir bases de conhecimentos, como Estudar o domínio da aplicação; determinar conceitos, objetos e relações necessários para resolução do problema.



4. O grafo a seguir apresentado representa um espaço de estados num problema de pesquisa, sendo D o estado objectivo (solução). Os custos das transições estão anotados junto às ligações do grafo e os valores da heurística estão anotados junto a cada nó.



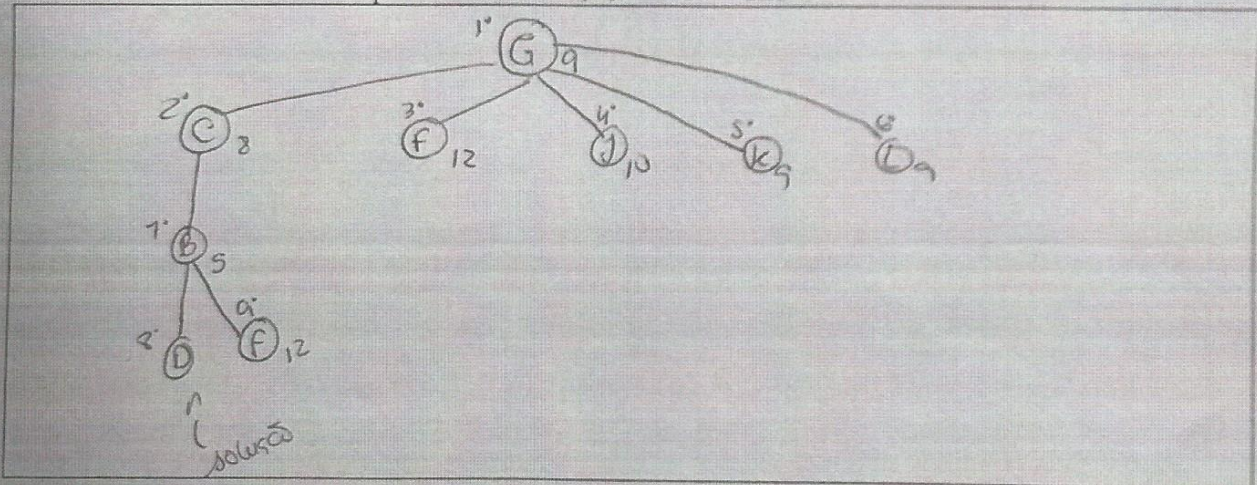
a) A heurística apresentada na figura é admissível? Justifique a sua resposta e, em caso negativo, faça as alterações necessárias por forma a que passe a sê-lo. [Pode fazer directamente na figura]

	D → A	D → B	D → C	D → E	D → F	D → G	D → H	D → I	D → J	D → K	D → L
Custo	5	6	8	10	11	12	8	5	10	9	13
Custo Estimado	5	5	8	10	12	9	6	4	10	9	9

x

Custo ≥ Custo Estimado      não é admissível, devido a F. (Trocar 7 por 11 ou valor maior)

b) Tomando o estado G como estado inicial, apresente a árvore de pesquisa gerada quando se realiza uma pesquisa gulosa. Esta pesquisa é feita sem repetição de estados no caminho de qualquer nó até à raiz da árvore. 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, deve desempatar com base na ordem alfabética dos respectivos estados. *sem valor alterado*



c) Calcule o factor de ramificação médio da árvore gerada.

$$RH = \frac{N-1}{x} = \frac{9-1}{3} = \frac{8}{3}$$

N = n total nós do árvore  
x = n nós expandidos