

Resolução Recurso 2015

I

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

a)

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

Fica assim

```
result = []  
for z in x:  
    if y(z):  
        result.append(z)  
  
return result != []
```

`y` é uma função que retorna um valor booleano

`x` é uma lista

R: Testa se algum elemento de `x`, transformado pela função `y`, devolve `True`.

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)
```

`y` é uma lista

`x` é um numero

A função devolve sempre um `tuplo(numero, lista)`

1. $g(4, [-1, 0, -3, 4, 2, 1]) (z, t) = g(4, [0, -3, 4, 2, 1]) \rightarrow$ passo 2
2. $g(4, [0, -3, 4, 2, 1]) (z, t) = g(4, [-3, 4, 2, 1]) \rightarrow$ passo 3
3. $g(4, [-3, 4, 2, 1]) (z, t) = g(4, [4, 2, 1]) \rightarrow$ passo 4
4. $g(4, [4, 2, 1]) (z, t) = g(4, [2, 1]) \rightarrow$ passo 5
5. $g(4, [2, 1]) (z, t) = g(4, [1]) \rightarrow$ passo 6
6. $g(4, [1]) (z, t) = g(4, []) \rightarrow$ passo 7
7. $g(4, []) (z, t) = \text{devolve } (0, [])$
8. $g(4, [-1, 0, -3, 4, 2, 1]) (z, t) = (1, [0, -3, 2, 1]) \text{ devolve } (1, [-1, 0, -3, 2, 1])$
9. $g(4, [0, -3, 4, 2, 1]) (z, t) = (1, [-3, 2, 1]), \text{ devolve } (1, [0, -3, 2, 1])$
10. $g(4, [-3, 4, 2, 1]) (z, t) = (1, [2, 1]), \text{ devolve } (1, [-3, 2, 1])$
11. $g(4, [4, 2, 1]) (z, t) = (0, [2, 1]), \text{ devolve } (1, [2, 1])$
12. $g(4, [2, 1]) (z, t) = (0, [1]), \text{ devolve } (0, [2, 1])$
13. $g(4, [1]) (z, t) = (0, []), \text{ devolve } (0, [1])$
14. $g(4, []) (z, t) = \text{devolve } (0, [])$

$g(4, [-1, 0, -3, 4, 2, 1]) = (1, [-1, 0, -3, 2, 1])$

R: Devolve um tuplo, cujo primeiro elemento é o número de vezes em que x está presente em y e cujo segundo elemento é y sem todas as ocorrências de x .

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

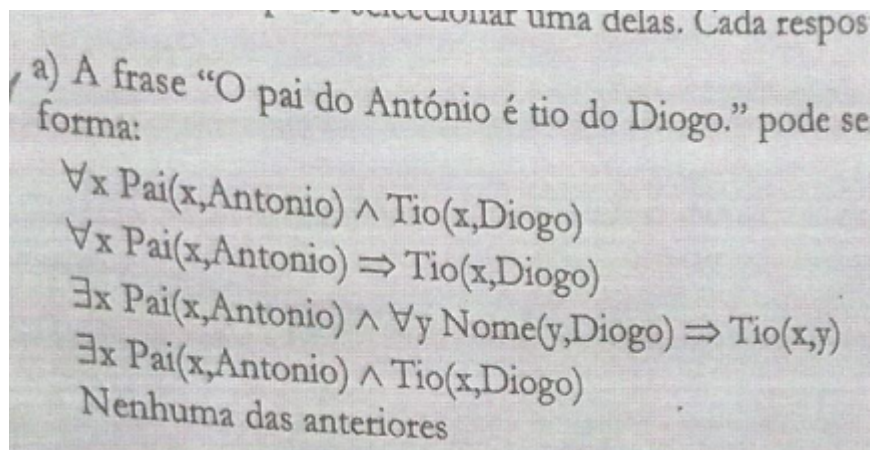
por modificação de uma dada solução; e uma função que avalia uma dada solução. Será valorizada a utilização de funções de ordem superior pré-definidas.

aitafoder

II

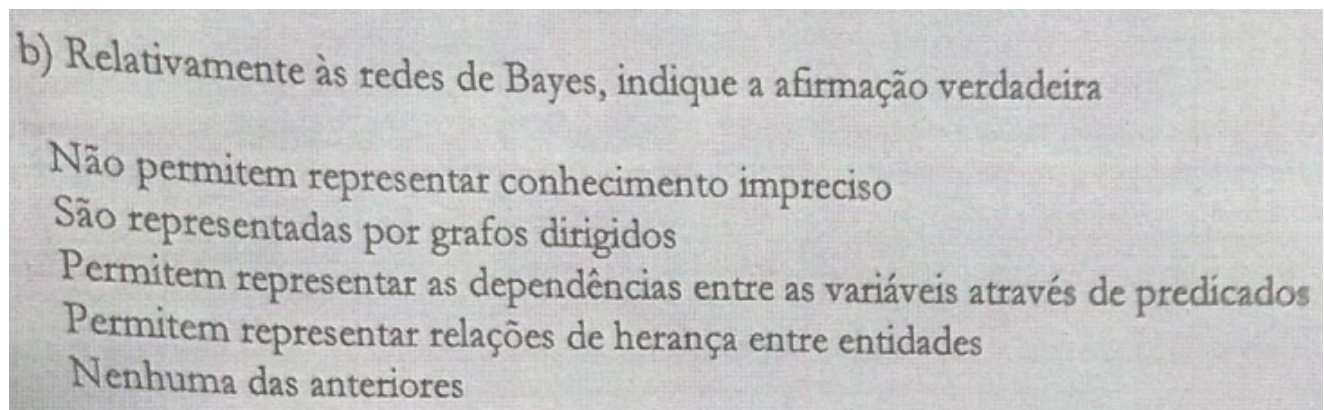
1.

a)



R: d)

b)



R: b)

c)

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

R: a)

d)

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

$\neg B \wedge C$

$A \vee D \vee C$

A

$A \vee \neg B$

Nenhuma das anteriores

R: e)

e)

✓ 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

R: a) ou d) se não for idkd, d se a) for sobre backtracking.

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 **deteta que o depósito está cheio**, **aguarda** que deixe de estar cheio, e só depois coloca lá a bola.

a) Identifique e caracterize as varias condições/predicados que podem ser usadas para descrever as situações em que se pode encontrar o robo. Identifique tambem quaisquer variaveis de estado necessarias.

Predicados:

- BolaFora(x)
- DepositoCheio()

Variaveis de estado:

- TemBola: boolean
- EmDeposito: boolean

b) Regras situação-ação que definam o comportamento do robo.

| Situação | Atualização | Ação |
|--|--|---------------------------------------|
| <code>!TemBola</code> | | Procura Bola |
| <code>BolaFora(x) ^ !TemBola</code> | <code>TemBola = True</code> | Agarra Bola |
| <code>TemBola ^ !EmDeposito</code> | <code>EmDeposito = True</code> | Deslocar para deposito |
| <code>TemBola ^ EmDeposito ^ DepositoCheio()</code> | | Aguardar ate deposito nao estar cheio |
| <code>TemBola ^ NoDeposito() ^ !DepositoCheio()</code> | <code>TemBola = False,</code> <code>EmDeposito = False</code> | Colocar a bola no deposito |

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

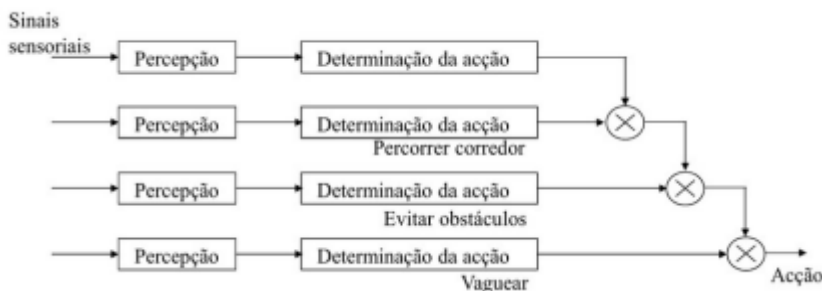
Existem agentes reativos e deliberativos. Os agentes reativos apresentam um *conjunto de sensores*, atraves dos quais recebe uma perceção do *estado do mundo*, sobre a qual aplica um *conjunto de regras* (**regras de condição-ação**) e executa as ações correspondentes com *atuadores*. - **Agente reativo simples**.

Os **agentes reativos com estado interno**, além de usarem sensores, fazem uso de um estado interno e do histórico de ações anteriores para construir a percepção do estado do mundo.

Os agentes deliberativos executam as ações com base em **objetivos** ou **função de utilidade**.

Uma das arquiteturas hierárquicas para agentes mais complexos é a **subsunção**: Esta arquitetura procura estabelecer a ligação entre a percepção e a ação em vários níveis, organizados em camadas, criando agentes com comportamento simultaneamente reativo e deliberativo.

A camada mais baixa é a mais reativa, diminuindo a reatividade e aumentando o peso da deliberação à medida que se sobe nas camadas. A ação do agente resulta da fusão das decisões das várias camadas.



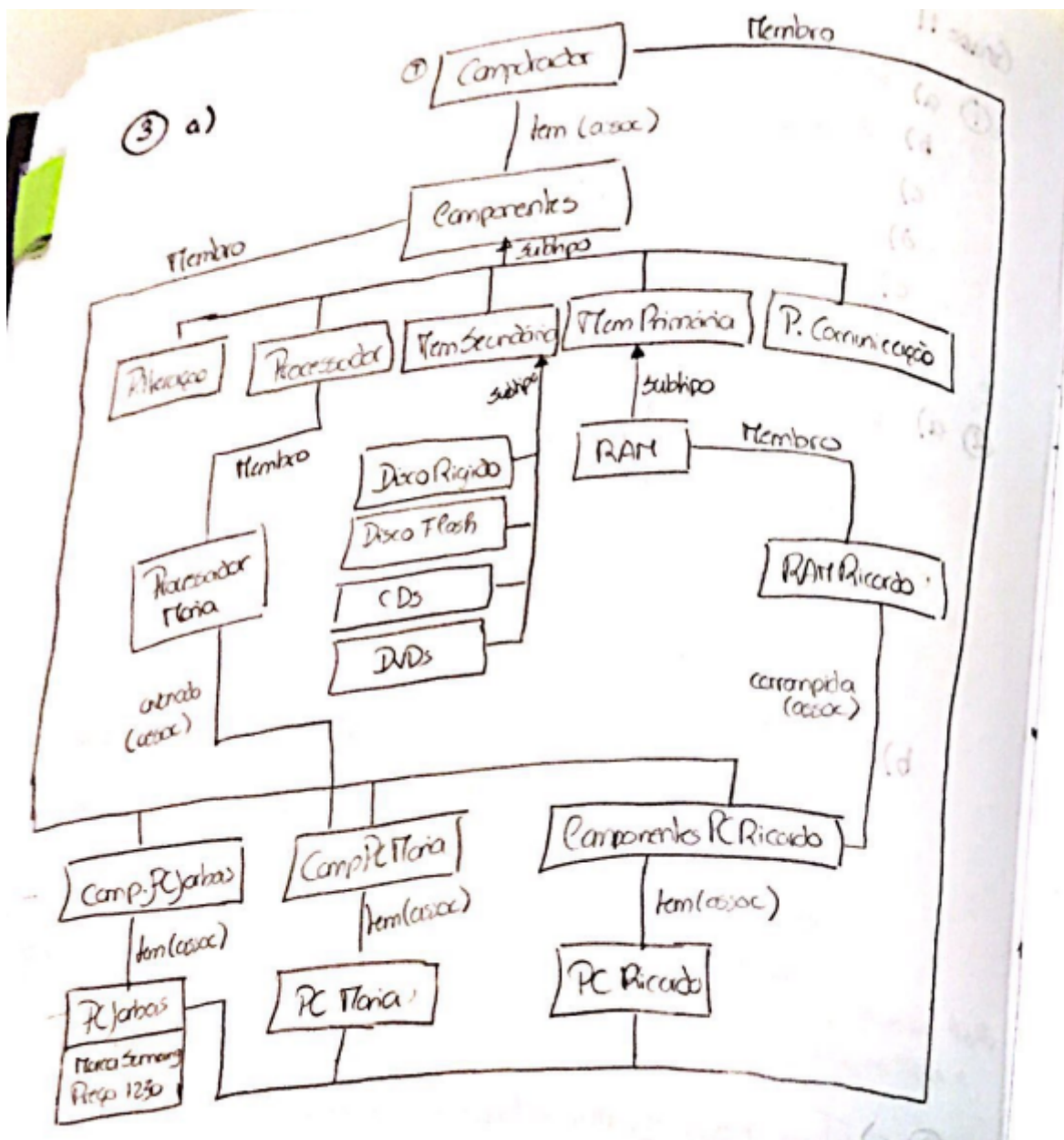
3. Um PC tem componentes: cpu, memória primária, memória secundária, periféricos de interação e periféricos de comunicação.

A Ram é um tipo de memória primária.

Os discos rígidos, discos flash, CDs e DVDs são diferentes formas de memória secundária.

- O pc do Ricardo tem a memória RAM corrompida.
- O computador da Maria tem o CPU avariado.
- O Jarbas comprou um pc Samsung por 1200 euros.

Represente esta informação através de uma rede semântica.

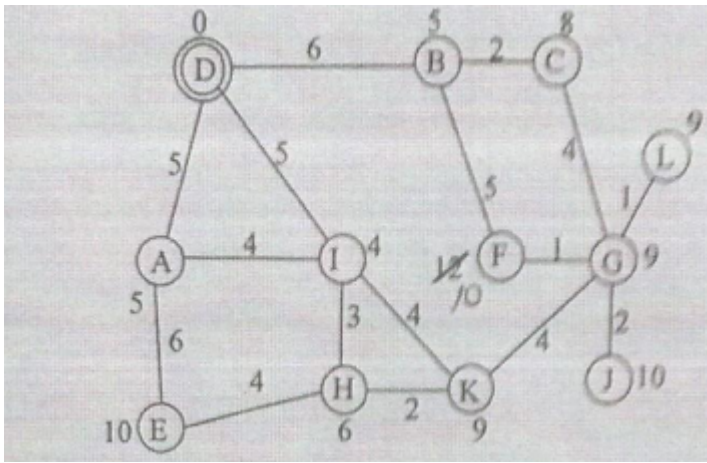


b) O que é Engenharia do Conhecimento

Uma base de conhecimento é um conjunto de representações de fatores e regras de funcionamento do mundo. Engenharia do conhecimento é o processo ou atividade de construir bases de conhecimentos:

- estudar o domínio da aplicação;
- determinar conceitos, objetos e relações necessárias para resolução do problema.

4. O grafo a seguir apresentado representa um espaço de estados num problema de pesquisa, sendo D o objetivo.



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]

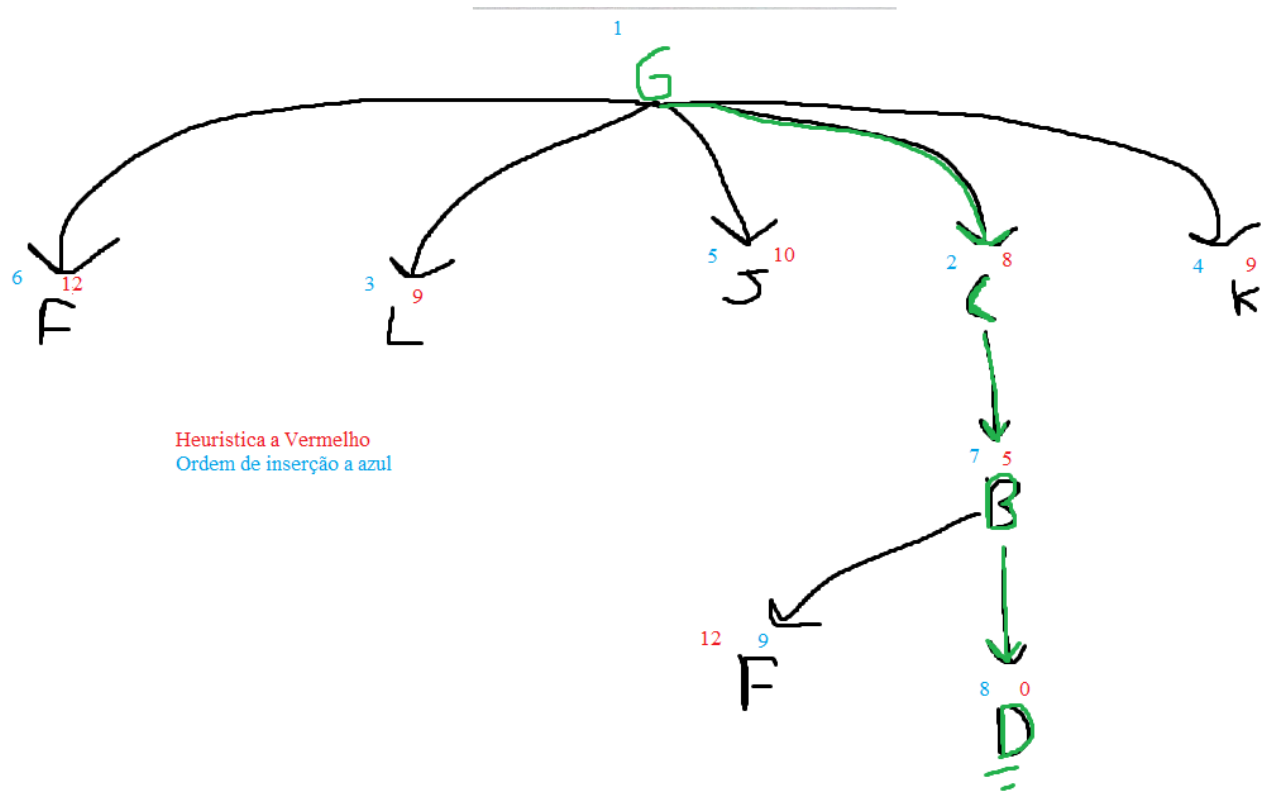
Temos que calcular o custo real de todos os nós até chegar ao objetivo. Para cada nó, temos que comparar esse valor obtido com a heurística do nó. Se a heurística *sobreestimar* o valor real, i.e $heurística > custo\ real$, a heurística não é admissível.

tem que ser $custo\ real \geq heurística$

| | Custo até D | Heurística |
|---|-------------|------------|
| A | 5 | 5 |
| B | 6 | 5 |
| C | 8 | 8 |
| E | 11 | 10 |
| F | 11 | 12 |
| G | 12 | 9 |
| H | 8 | 6 |
| I | 5 | 4 |
| J | 14 | 10 |
| K | 9 | 9 |
| L | 13 | 9 |

Em F, não é admissível, temos que baixar a heurística uma unidade.

b) Pesquisa Gulosa no grafo, sabendo que começamos em G. Apresentar ordem de inserção e anotar o valor de função em cada nó



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

ramificação média indica nos a dificuldade do problema

- N - número de nós da árvore de pesquisa no momento em que se encontra a solução
- X - número de nós expandidos (não terminais)

$$RM = \frac{N - 1}{X} = \frac{9 - 1}{3} = 2.67$$