

# **Inteligência Artificial**

---

Resumos  
2017/2018

Carolina Albuquerque | 80038

# Inteligência Artificial



## Resumos Exame Teórico

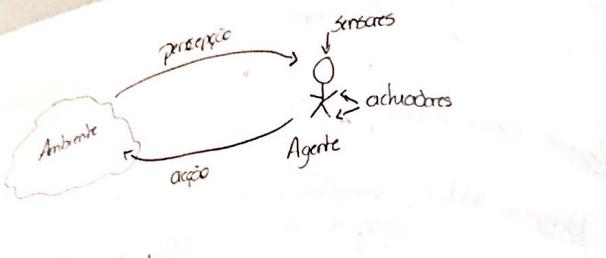
Inteligência Artificial - disciplina que estuda as teorias e técnicas necessárias ao desenvolvimento de "artefactos" inteligentes

### Agenres

→ Entidade com capacidade de obter informação sobre o seu ambiente (através de "sensores") e de executar ações em função dessa informação (através de "actuadores")

### Propriedades do Mundo de um Agente:

- Acessibilidade - se os "sensores" do agente permitem obter uma descrição completa do estado do mundo, o mundo é efectivamente acessível se for possível a obtenção de toda a informação relevante ao processo de escolha das ações.
- Determinismo - se o estado resultante da execução de uma ação é totalmente determinado pelo estado atual e efeitos esperados da ação
- Mundo Episódico - no caso em que cada episódio de percepção-ação é totalmente independente dos outros
- Dinamismo - se o seu estado pode mudar enquanto o agente aíbera, caso contrário é estático
- Continuidade - quando a evolução do estado do mundo é processo contínuo ou sem saltos, caso contrário é discreto



### Exemplos de Agentes

Tipo Agente	Percepção	Ação	Objetivos	Ambiente
Sistema Diagnóstico TID	Sintomas, Respostas do paciente	Tratamentos, Testes	Saúde Paciente	Paciente, Hospital
Tutor Inglês Iterativo	Palavras Traduzidas	Propri. ex., config.-pos	Maximizar o resultado alunos	Projunto de alunos

Comportamento Inteligente - capacidade de um artefato obter desempenho comparável ao desempenho humano em todas as atividades cognitivas

Teste de Turing - é uma definição operacional de comportamento inteligente de nível humano

- consiste em submeter o artefato a um interrogatório realizado por um ser humano através de um terminal texto
- Se o humano não consegue concluir se está a interrogar um artefato ou um ser humano, então o artefato é inteligente

### Tipos e Arquiteturas de Agentes

Tipos de Agentes

- Reativos Simples
- Reativos com estrada
- Deliberativos orientados por objetos
- Deliberativos orientados por funções de utilidades

Arquiteturas

- Substituição
- Três Torres
- Três Caixas
- CARL

### Redes Semânticas

- > Representações gráficas do conhecimento e sua principal vantagem é a legibilidade
- > Podem ser tão expressivos como a Lógica de PrIMEIra Ordem
- > Permitir representar conhecimento por omissão

### Tipos de Relação:

Subtipo  $A \subset B$

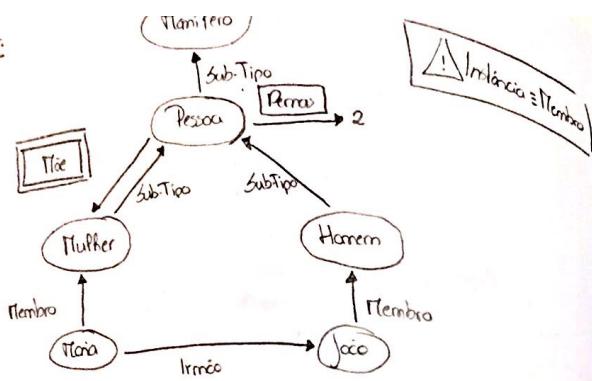
Membro  $A \in B$

Relação Objeto-Objeto  $R(A, B)$

Relação Conjunto-Objeto  $\forall x, x \in A \Rightarrow R(x, B)$

Relação Conjunto-Conjunto  $\forall x, x \in A \Rightarrow \exists y (y \in B \wedge R(x, y))$

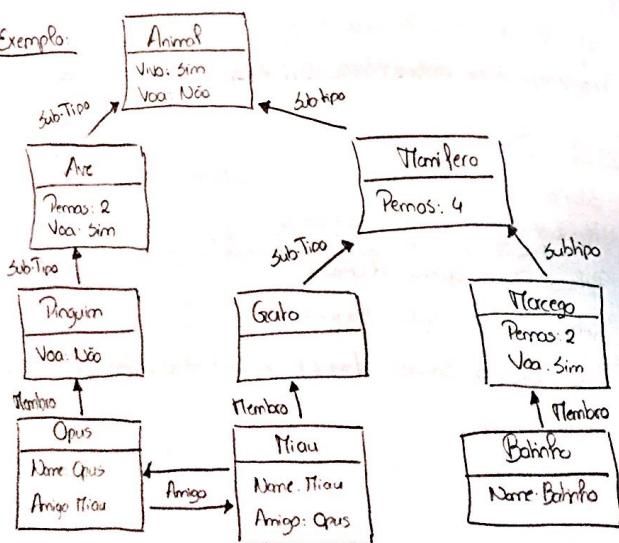
Exemplo:



> As relações de sub-tipo e instância permitem:

- sub tipo herda todas as propriedades dos tipos mais abstratos dos quais descendem
- instância herda todas as propriedades do tipo a que pertence

Exemplo:



**⚠️ Nota:** Observar-se redes semânticas muito menos expressivas em que a Lógica de 1º ordem (razões computacionais)

Dedução	Indução
> Permite inferir casos particulares a regras gerais	> Oposto da dedução, permite inferir regras gerais através de casos particulares
> Preserva a verdade	> Regras de inferência são drátilas
> Base principal da aprendizagem	

### Redes Semânticas em Python

**Relação** {
 

- Member (Obj, Type)
- SubType (SubType, SuperType)
- Association (Entity1, AssocName, Entity2)

}

**Operações Principais:** insert (introduzir uma nova declaração) e query\_local (questionar a rede semântica sobre os de declarações existentes)

### KIF

- > Linguagem desenhada para representar o conhecimento tratado entre agentes
- > Pode ser usada também para representar os modelos internos de cada agente

#### Características Principais:

- > Pode ser tão ou mais extensa que a Lógica de primeiro ordem
- > Permite a representação de meta-conhecimento conhecimento sobre conhecimento
- > Relação conjunta arbitrária de listas objetos

### Emoção do Conhecimento

- > Base & conhecimento - conjunto de representações & factores e regras & fundamentos do mundo
- > Emoção do Conhecimento - processo ou atividade de construir bases & conhecimento: estudar o domínio da aplicação, definir conceitos, objectos, relações que se querem representar, codificar o conhecimento genérico sobre o domínio

### Redes Bayes vs Redes Semânticas

Rede Semântica	Rede Bayes
Baseadas em Relações	Baseadas em Probabilidades
Não pode ser tipo ou objeto	Um nó representa uma dependência prob
Gráficos Dirigidos	

### Exame Exemplo 1A

#### Grupo II

- ① a) Nenhuma das anteriores (5º opção)
- b) Vai Ativo( $\pi$ )  $\Rightarrow$  (Nota(Ara, Português)  $\geq$  Nota(Lu, Português)) (4º opção)
- c) Técnica de pesquisa para diminuir soluções (3º opção)
- d) C (2º opção)
- e) Nenhuma das Anteriores (5º opção)

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

Página 148 PPT

#### Diferenças:

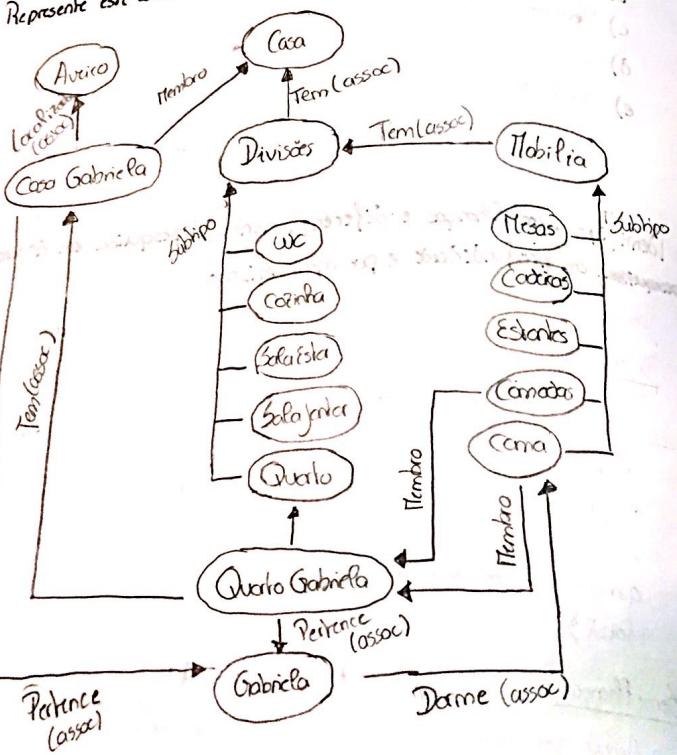
- > No Montanhismo escolhe-se sempre o sucessor com melhor valor da função de avaliação
- > Não há retrocesso no montanhismo
- > Quando o valor da função no nó atual é superior ao valor da função em qualquer um dos seus sucessores, a pesquisa pára. (É chegado um máximo local).

#### Semelhanças:

- > Ambas permitem encontrar o caminho de menor custo para o objetivo

③ As casas têm divisões de diferentes tipos, por exemplo, salas de estar, salas jantar, quartos dormir, cozinhas e quartos banho. As divisões da casa têm tipos mobiliário, como por exemplo, mesas, cadeiras, camas, cãmodas, estantes. A casa da Gabriela é em Aveiro. Essa casa tem um quarto com uma cama, em que a Gabriela dorme, e uma cômoda.

Represente este conhecimento através de uma rede semântica.



Casa - Type

Divisão - Type

SubType (Sala Estar, Divisão)

SubType (Sala Jantar, Divisão)

SubType (Cozinha, Divisão)

SubType (WC, Divisão)

SubType (Quarto, Divisão)

Mobilíario - Type

SubType (Mesa, Mobilíario)

SubType (Cadeira, Mobilíario)

SubType (Estante, Mobilíario)

SubType (Cômoda, Mobilíario)

SubType (Estante, Mobilíario)

Member (Perna Gabriela, Perna)

Member (Cômoda Gabriela, Cômoda)

Member (Quarto Gabriela, Quarto)

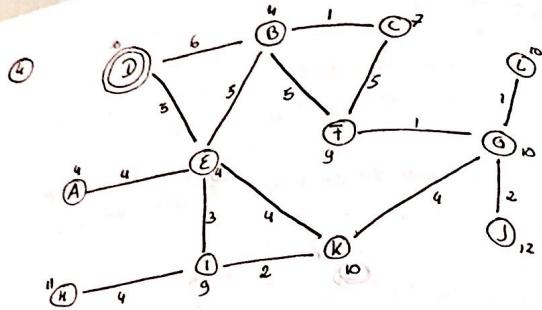
Assoc (Quarto Gabriela, Item, Cama Gabriela)

Assoc (Quarto Gabriela, Item, Cômoda Gabriela)

Member (Casa Gabriela, Casa)

Assoc (Casa Gabriela, localizada, Aveiro)

Assoc (Casa Gabriela, Item, Quarto (Gabriela))



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 alterações que tornem admissível. Justifique.

⚠ Pesquisa A\*:

$$f(n) = g(n) + h(n)$$

$g(n)$  - custo des de nó inicial ate ao n

$h(n)$  - custo estimado des de o n ate a solução

Nó A:  $g(n) = 9$   
 $h(n) = 4$

Nó H:  $g(n) = 12$   
 $h(n) = 11$

Nó B:  $g(n) = 6$   
 $h(n) = 4$

Nó I:  $g(n) = 8$   
 $h(n) = 9$

Nó C:  $g(n) = 7$   
 $h(n) = 4$

Nó K:  $g(n) = 9$   
 $h(n) = 10$

Nó E:  $g(n) = 5$   
 $h(n) = 4$

Nó J:  $g(n) = 14$   
 $h(n) = 12$

∴ Nó E é admissível porque procura K e I, o  $h(n)$  é superior a  $g(n)$ .

Nó F:  $g(n) = 11$   
 $h(n) = 9$

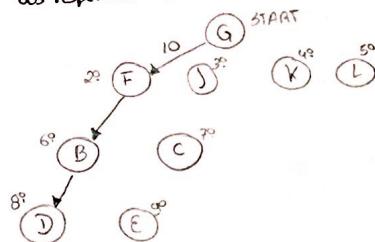
Nó L:  $g(n) = 13$   
 $h(n) = 10$

Alterações: baixa o  $h(n)$  em ambos os casos de modo  $h(n) \leq g(n)$

Nó G:  $g(n) = 12$   
 $h(n) = 10$

⚠ Nota:  $h(n)$  é admissível  
 Sempre sobreestima o custo real de chegar a uma solução a partir de n  
 ↓  
 A\* ótima

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 apresentados à árvore e entre tanto 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.



$$\begin{aligned} & f(G, F) = 1 + 9 = 10 \\ & f(G, J) = 2 + 12 = 14 \\ & f(G, C) = 4 + 10 = 14 \\ & f(G, B) = 1 + 10 = 11 \\ & f(F, D) = 4 + 6 = 10 \\ & f(F, E) = 5 + 8 = 13 \\ & f(B, D) = 0 + 6 = 6 \\ & f(B, E) = 4 + 5 = 9 \end{aligned}$$

Ordem Pesquisa: G - F - J - K - L - B - C - D - E

⚠ Letras minúsculas

Parce(N, P) - no N tem parceiro P

Estracionado(P) - estracionado em P,

Em(N, N2) - está no nó N do lado de N2

Rua(N1, N2) - existe uma rua que liga N1 e N2

b) Estracionado(N, N2, P)

Pré-Condições: Em(N1, N2), Parce(N1, P)

Efeitos Negativos: Em(N1, N2)

Efeitos Positivos: Estracionado(P)

Atravessar(N1, N2, N3)

Pré-Condição: Rua(N1, N2), Rua(N2, N3), Em(N1, N2)

Efeitos Negativos: Em(N1, N2)

Efeitos Positivos: Em(N2, N3)

Sair ( $n_1, n_2$ )  
 Efeitos: Estacionando ( $P$ ), Rua ( $n_1, n_2$ ), Roubo ( $n_1, p$ )  
 Efeitos Negativos: Estacionando ( $P$ )  
 Efeitos Positivos: Em ( $n_1, p_2$ )  
 Parar ( $n_1, n_2$ )  
 Efeitos: Em ( $n_1, n_2$ ), Rua ( $n_1, n_2$ )  
 Efeitos Negativos: Em ( $n_1, n_2$ )  
 Efeitos Positivos: Em ( $n_2, n_1$ )

### Redes de Bayes

- > Permitem representar o conhecimento empírico em termos de um conjunto de variáveis aleatórias e respectivas dependências.
- > As dependências são expressas através de probabilidades condicionadas
- > Rede é um grafo dirigido acíclico.

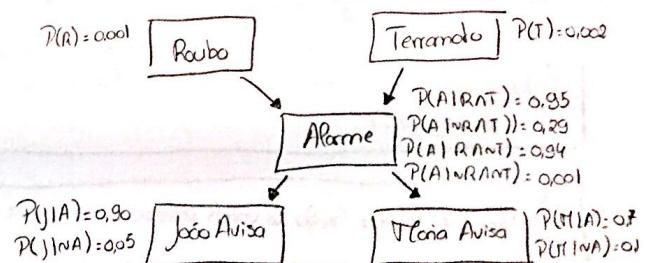
Exemplo: Situação: Sistema Alarme Instalado em Casa

- > Ativado em situações de Roubo
- > Ativado em humor de Terra

Dois vizinhos que prometem avisar quando o ofício é acionado.

- > Tloia: ouve música alta e pode não ouvir alarme
- > Jóia: telefone sempre que ouve o alarme mas confunde-o muitas vezes

Questão: Dada a evidência, qual a probabilidade de ocorrer um assalto?

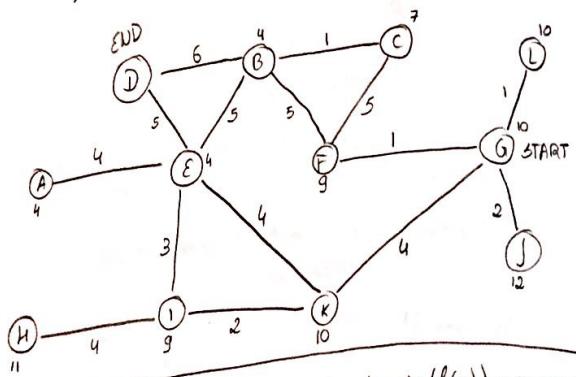


Roubo	Término
V	V
V	F
F	V
F	F

É necessário especificar a tabela de Probabilidades Condicionadas para cada nó

### Pesquisa em Averro

- > Pesquisa em Largura (Breadth-Search) - abri todos os nós e percorre os perniveis
- > Pesquisa em Profundidade (Depth-First Search) - custo real, sem repetição de estados ou com repetição de estados
- > Pesquisa Greedy (Greedy) - custo estimado ( $h(n)$ )
- > Pesquisa em Custo Uniforme - custo acumulado ( $g(n)$ )
- > A\* - custo acumulado + custo estimado ( $h(n) + g(n)$ )



**⚠️** Nota: Junto a cada nó temos o custo estimado ( $h(n)$ ) e no caminho o custo do nó ( $g(n)$ )

**⚠️** Vamos Aplicar o Grafo as várias pesquisas em árvore

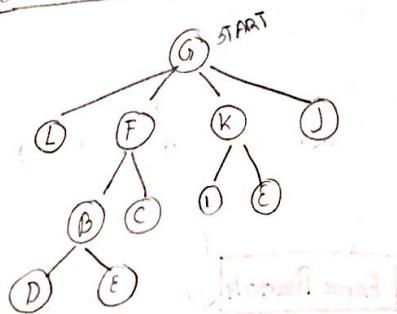
### Pesquisa em Largura:

- > Começamos em G e vamos ver todos os que estão ligados a G
- G - K - F - L - J - E - I - B - C - D

### Pesquisa em Profundidade:

- > G - F - B - D

### Pesquisa Custo Uniforme:

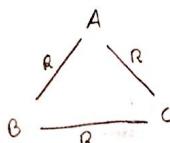


Ordem Pesquisa: G - F - J - K - L - B - C - I - E - D

*... os meus três amigos voltaram*

$A, B, C$  - são objetos cada um  
 $A \in \{bb, cc\}, \cancel{\{bb, bc\}} \rightarrow$  Restrição unária

$$\left. \begin{array}{l} A \in \{(bb,cc), \\ (cc,cc)\} \\ B \in \{(ba,cc), \\ (ca,cc)\} \\ C \in \{(ba,cb), \\ (cb,cc)\} \end{array} \right\} \text{As sublinhados enunciado}$$



$$B_{(x,y)} = (x[0] \neq y[0] \wedge x[1] \neq y[1])$$

$$R(x, v_x, y, v_y) = (v_x[0] \neq v_y[0] \wedge v_x[1] \neq v_y[1])$$

## Exercício Recurso 14

① a) Verifica se um elemento de  $\mathbb{N}$  satisfaz a condição  $y(n)$

b)  $x=2$  } Verifica se o elemento  $n$  está na lista  $y$  e devolve um tuplo  
 $y=[1, 2, 3]$  } com contagem de vezes que esse elemento aparece na lista e  
                   a lista sem esse elemento

② def. get.ancestors (self, lista, var):

```

if vor not in fisba:
    fisba.add(vor)
    for parents in selff.dependencies[vor]:
        for pver, pval in parents:
            selff.ancestors(pver, r)
return r

```

## Grupo II

- ① a)  $4^{\circ}$  φέγγο  
b)  $2^{\circ}$  φέγγο  
c)  $1^{\circ}$  φέγγο  
d)  $24^{\circ}$  φέγγο  
e)  $40^{\circ}$  φέγγο

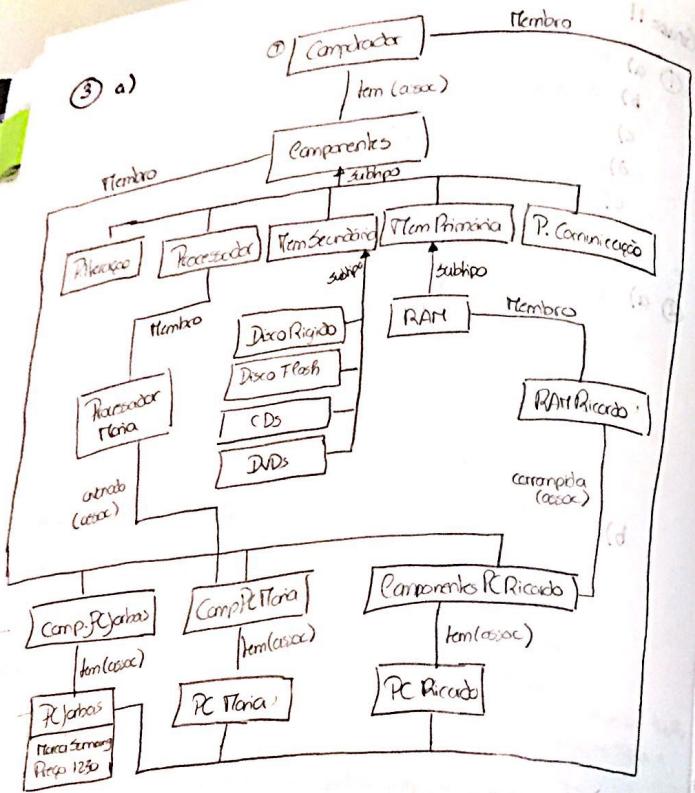
② a) Estados f. Predicado:

Bela(x) - o robô tembra bela x  
 DepositoChoco - variável do estado  
 BelaTara(x) - Bela x fera de campo  
 BelaBaliza(x) - " " na baliza.

Situação	Atribuição	Acão
$Bela(x) \wedge Bela(f(x)) \wedge ^*Bela(y)$	—	Agrava(x)
$(^*Bela(x) \wedge ^*Belaf(f(x)) \wedge ^*Bela(z))$	—	Procura( )
$Bela(x) \wedge ^*DepositoCheio$	—	Deposito(x)
$Bela(x) \wedge DepositoCheio$	—	Especula( )

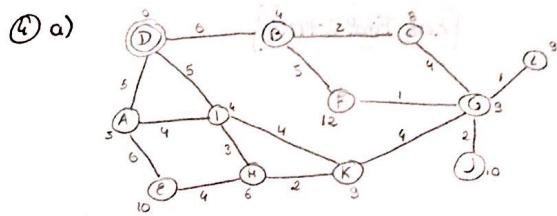
c) Tipos de agentes: reativos (aqueles que reagem por imediato a algo) e deliberativos (aqueles que fazem pesquisas / cálculos)

### Akavitras:



b) A engenharia do conhecimento é o processo ou atividade de construir bases de conhecimento (conjunto de representações e regras para o funcionamento do mundo) envolvendo:

- estudo domínio e aplicação
- definir um vocabulário para entidades, funções, relações
- determinar os objetos, conceitos, relações que serão necessários representar
- codificar conhecimento genérico sobre o domínio
- (...)



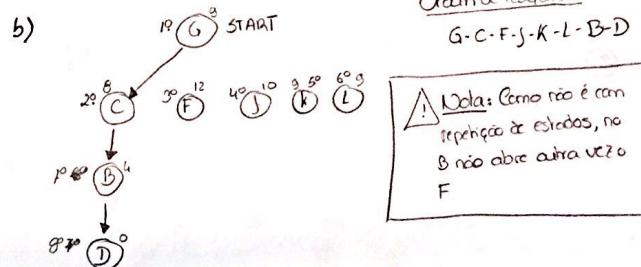
$g(n) \rightarrow$  custo real

$f(n) \rightarrow$  custo estimado

Nós	A	B	C	D	E	F	G	H	I	J	K	L
Custo Real	5	6	8		11	11	12	8	3	13	9	12
Custo Estimado	5	4	8		10	12	9	6	4	10	9	9

A heurística não é admissível porque partindo do nó F, o custo estimado é maior que o custo real.

Alteração: Baixa o custo estimado do nó para 10.



c) Taxa Ramificação =  $\frac{N \cdot 1}{n}$ ,  $N$  - nº total nós abertos no momento da simulação  
 $n$  - nº nós abertos

$$= \frac{f+1}{3} = 2$$

Linha

① a) Sabemos que  $x$  é um elemento,  $y$  é uma lista e  $z$  é uma função.

$$x = 1$$

$$y = [1, 2, 3]$$

Inserção do elemento  $x$  na array  $y$ , assim que se satisfaz a  $z(\dots)$

b)  $x$  e  $y$  são listas e  $z$  é uma função

Junta as duas listas.

Exemplo:

$$x = [1, 2, 3] \quad \{ \quad 1, 2, 3, 7, 5, 4$$

$$y = [7, 5, 4]$$

$$x = [1, 2, 3] \quad \{ \quad 1, 2, 3, 4, 5, 6, 4$$

$$y = [2, 3, 6]$$

②

Grupo II

① a) 2<sup>a</sup> opção

b) 3<sup>a</sup> opção

c) 4<sup>a</sup> opção

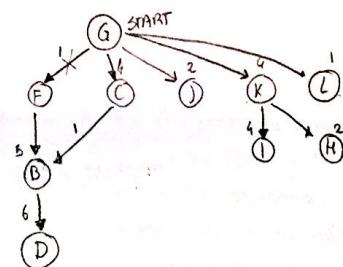
d) 5<sup>a</sup> opção

e) 3<sup>a</sup> opção

② a) Deteta ( $n$ ),  $n \in \{\text{aqui, trás, frenar, esquerda, direita}\}$ . Deteta fixo em se Tempo - variável

b)	Situação	A	Atribuição	Ação
	Deteta(Frenar)	—	—	Averga () $\wedge$ Aspira ()
	Deteta(Trás)	—	—	Roda (90°) $\wedge$ Roda (90°) $\wedge$
	Deteta(Direita)	—	—	Averga e aspira
	Deteta(Esquerda)	—	—	Roda (90°) $\wedge$ Averga $\wedge$ Aspira
	Deteta( $n$ ) $\wedge$ Temporais	—	—	(Roda (90°)) $\times 3$ , Averga $\wedge$ Aspira

③ a)



$$Tm = \frac{N-1}{n} = \frac{9}{5}$$

Odm Pesquisa: G - C - F - J - K - L - B - I - H - D

X Averg Pesquisa Incompleta

## Guia Teórico Prático

### Grupo II

- ① a) ArpoesDisponíveis = 10  
CapDepósito = 20
- b) Deteta(x) - o robô detecta um peixe x  
Atingido(x) - o peixe x foi atingido  
Guardado(x) - o peixe x foi guardado  
Tanque() - tanque está cheio  
Arpoes() - ficou sem arpoes

Situação	Atualização	Ação
Deteta(x) $\wedge$ Arpoes()	ArpoesDisp --	
Arpoes()	ArpoesDisp = 10	Disparar
Tanque()	CapDepósito = 20	Abastecer
Atingido(x) $\wedge$ Deteta(x)	CapDepósito = 1	Desarranjar
$\neg$ Deteta(x) $\wedge$ Tanque() $\wedge$ Arpoes() $\wedge$ Atingido()	ArpoesDisp += 1	Agarrar
	—	Vaguear

- ② a) distância - distância percorrida pela formiga
- b) Encontra(x) - é encontrada uma provisão x  
Possui(x) - a formiga tem na sua posse a provisão x  
Vê(y) - a formiga vê a formiga y  
Local(l) - a formiga encontra um local l

Situação	Atualização	Ação
Encontra(x) $\wedge$ Possui(x)	distância = 0	AgarrarProvisão(x)
$\neg$ Encontra(x) $\wedge$ Possui(x)	distância +=	ProcurarProvisão(x)
Possui(x) $\wedge$ distância > 5	—	ProcurarLocal(l)
$\wedge$ Local(l)	—	—
Possui(x) $\wedge$ distância > 5	—	SeguirFormiga(y)
$\wedge$ Local(l) $\wedge$ Vê(y)	—	LiberarProvisão(x)
Possui(x) $\wedge$ Local(l)	—	—

### Grupo III

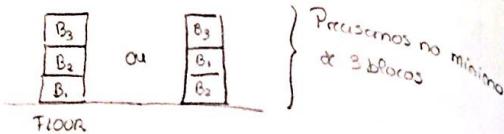
- ① a)  $\forall x$  Oxford(x)  $\Rightarrow$  Esperito(x)
- b)  $\exists x$  Oxford(x)  $\wedge$  Esperito(x)
- c)  $\exists x \forall y$  Gosta(x,y)
- d)  $\exists y$  chumbou(y, História)  $\wedge$   $\forall x$   $\neg$  chumbou(x, História),  $\neg y$  História uma constante
- e)  $\forall x$   $\neg$  inscreve(x,IA)  $\wedge$  inscreve(x,so)
- f)  $\exists y$  [chumbou(x,História)  $\wedge$  chumbou(x,Biologia)]  $\wedge$   $\forall z$  [ $\neg$  chumbou(z,História)  $\vee$  chumbou(z,Biologia)],  $\neg x \neg y$
- g)  $\exists x \forall y$  Nota(x,História)  $>$  Nota(y,Biologia)
- h)  $\forall x$  Pintista(x)  $\Rightarrow$  Gosta(x,Pintor da Costa)
- i)  $\exists y$  Enthousiasta(y)  $\wedge$   $\forall x$  [Enthousiasta(x)  $\wedge$  Esperito(x)],  $\neg x \neg y$

② a)  $(\text{clear}(B_1)) \wedge (\text{clear}(B_2)) \equiv ?$  ( $\text{clear}(B_1) \vee (\text{clear}(B_2))$ )

Bloco  $B_3$  está em cima de outro bloco que não toca no chão

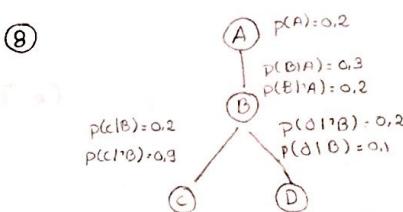
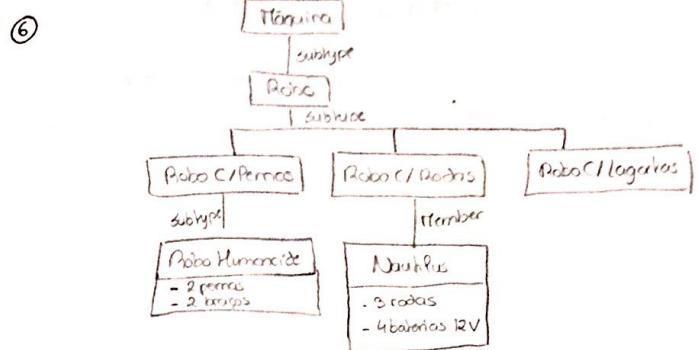
Bloco  $B_1$  tem um bloco por cima

Bloco  $B_2$  tem um bloco por cima



b)  $\boxed{B_1} \quad \boxed{B_2}$  ou  $\boxed{\begin{matrix} B_1 \\ B_2 \end{matrix}}$  } Pressionos no mínimo 2 blocos

③ a)



$$\begin{aligned}
 & p(a \wedge b \wedge c \wedge \neg d) \\
 & = p(b|a) \times p(c|b) \times p(d|c) \times p(\neg d|a) \\
 & = 0,3 \times 0,8 \times 0,9 \times 0,2 \\
 & p(\neg c|B) = 1 - 0,2 = 0,8 \\
 & p(\neg d|B) = 1 - 0,1 = 0,9
 \end{aligned}$$

- ⑯ - 60% utilizadores têm sobrecarga tridimensional → causa (pode) cora preocupada  
- 1% utilizadores cora preocupada | caso não precisam ajuda  
- 2% | caso precisam ajuda  
- 0,1% utilizadores não sobrecarregados, não precisam ajuda e mostram cora preocupada  
- 1,1% " " " " precisam ajuda, " " " "

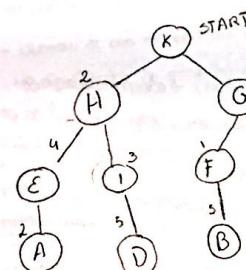
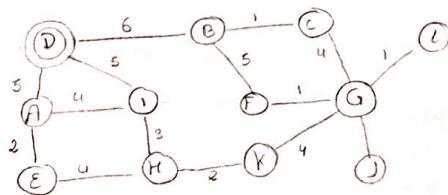
$P(b) = 0.60$   
 $P(\neg b) = 0.4$   
 $P(A|b) = 0.9$   
 $P(A|\neg b) = 0.1$

// Probabilidade de haver sobrecarga  
 // " " " não " "  
 // " " sem carro acarretado sabendo que haver  
 // sobrecarga de trabalho  
 // Probabilidade de acarretarem carros sem sobrecarga trabalho

Exame CT 2014/2015

① a)  $m$  é uma lista e y

③ a)



⚠ Árvore Pesquisa Incompleta

### Técnicas Busca Problemas

- ① Técnicas Pesquisa em Árvore
- ② Técnicas Pesquisa em Grafo
- ③ Técnicas Pesquisa Itinerários Sucessivos
- ④ Técnicas Pesquisa Propagação Restrições
- ⑤ Técnicas Planejamento

### ① Pesquisa Árvore

#### > Estratégias Pesquisa Grafo (Não informada)

Em largura  
Ciclo Uniforme ( $g(n)$ )  
Em profundidade  
Em profundidade com limite  
Em profundidade com limite superior

#### > Estratégias Pesquisa Informada

Pesquisa Gulosa ( $g(n)$ ) ( $h(n)$ )  
Pesquisa A\* (custo variável) ( $f(n) = g(n) + h(n)$ )

### ② Algoritmo Genérico

Pesquisa (Problema, Estratégia) → retorna solução ou "fallou"  
Árvore - inicializada com o estado inicial do problema

Pesquisa em Largura (Problema)  
retorna pesquisa em árvore (Problema, juntar.no.Prim)

Pesquisa em Profundidade (Problema)  
retorna pesquisa em árvore (Problema, juntar.círculo)

### Avaliação das Estratégias Pesquisas

- > Completeness - uma estratégia é completa se é capaz de encontrar uma solução quando existe uma solução
- > Complexidade Temporal - quanto tempo demora a encontrar a solução
- > Complexidade Espacial - quanto o espaço de memória é necessário para encontrar uma solução
- > Optimalidade - a estratégia de pesquisa consegue encontrar a melhor solução

### Fator Ramificação

> Avaliação da pesquisa em árvore

Ramificação Média:

$$RM = \frac{N-1}{n}$$

Fator Ramificação Efetivo:

$$\frac{B^{d+1} - 1}{B - 1} = N$$

N - número nos nós de pesquisa no momento em que se encontra a solução

d - número de nós expandidos (não terminais)

δ - comprimento do caminho na árvore correspondente à solução

#### Notas:

A RM é um indicador da dificuldade do problema

#### Notas:

Fator Ramificação Efetiva indicação de eficiência da técnica de pesquisa

### Vantagens e Desvantagens Pesquisa A\*

Vantagens: Completa e Ótima

Desvantagens: O consumo memória e tempo de computação têm um comportamento exponencial em função do tamanho da solução  
em problemas ④ complexos poderá ser preciso usar algoritmos mais eficientes.

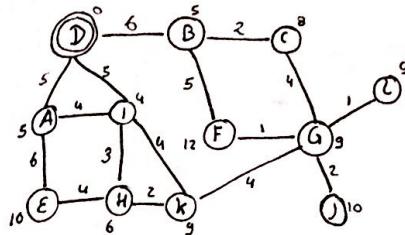
uso de heurísticas com uma melhor aproximação não garantem a otimização da pesquisa

### Técnicas Pesquisa Grafos

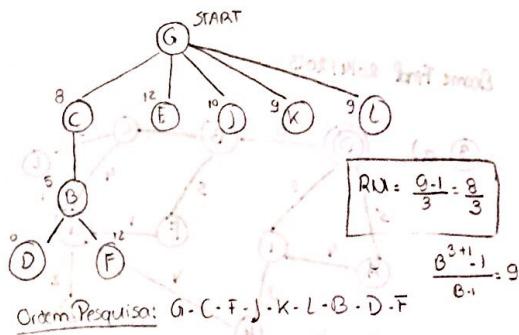
- Estratégia Pesquisa Largura (Breadth First Search)
  - {
  - Estratégia Pesquisa Largura (BFS)
  - Estratégia Pesquisa Uniforme em profundidade (Uniform Depth Search)
- Pesquisa em Árvore
  - {
  - Pesquisa Uniforme
  - Pesquisa Primitiva
  - Pesquisa Greedy (Greedy) , A\*
- Pesquisa em Grafo
  - {
  - Itinerário
  - Recorrendo Simples
  - Algoritmo Genéricos
- Pesquisa Multíplos Sucessores
  - {
  - Recorrendo Simples
- Pesquisa Propagação Restrições
- Planejamento STRIPS

### Correção dos Grafos

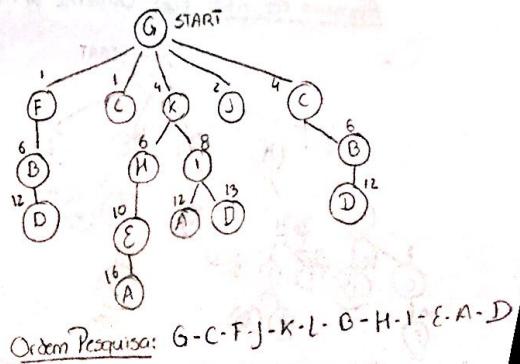
Pesquisa 2014/2015



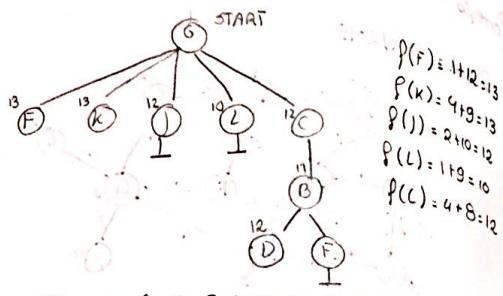
Pesquisa: Greedy sem repetição estados



Pesquisa: Uniforme sem repetição estados

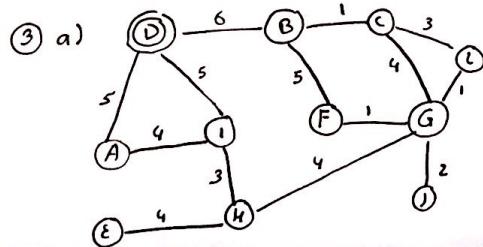


Pesquisa:  $A^*$  sem Repetição

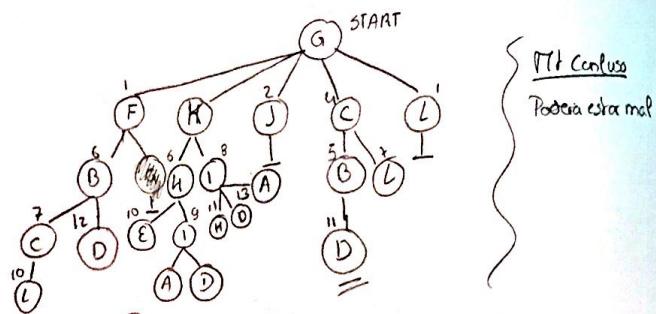


Ordem Pesquisa: G - C - F - J - K - L - B - D

Exame Find 2014/2015



Algoritmo Pesquisa: Custo Uniforme se não houver estados



Ordem Pesquisa: G - F - C - H - J - L - B - D  
G - C - F - J - K - L - B - C - B

Operadores STRIPS

Exemplo: Agarrar(x):

Pré-condição: ( $\neg$  Tem(x)  $\wedge$  Boliza(x))  $\vee$  ( $\neg$  Tem(x)  $\wedge$  Foco(x))  $\times$  Não são simultâneos

Efeito Negativo: Boliza(x)  $\wedge$  Foco(x)

Efeito Positivo: Tem(x)

SEPARAR PREDICADOS

AgarrarBoliza(x) - agarra na boliza a bala x

Melio Campo - vence  
se true ou false

AgarrarFoco(x) - agarra pega a bala x

AgarrarBoliza(x):

Pré-condição:  $\neg$  Tem(x)  $\wedge$  Boliza(x)

Efeito Positivo: Tem(x)

Efeito Negativo: Boliza(x)

AgarrarFoco(x):

Pré-condição:  $\neg$  Tem(x)  $\wedge$  Foco(x)

Efeito Positivo: Tem(x)

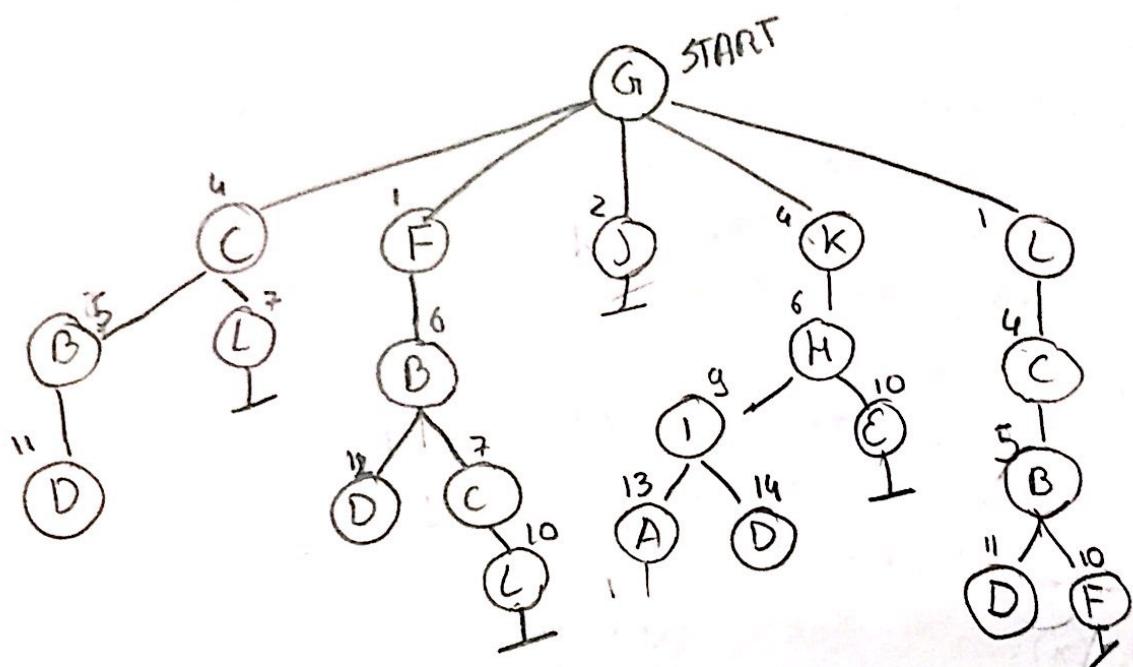
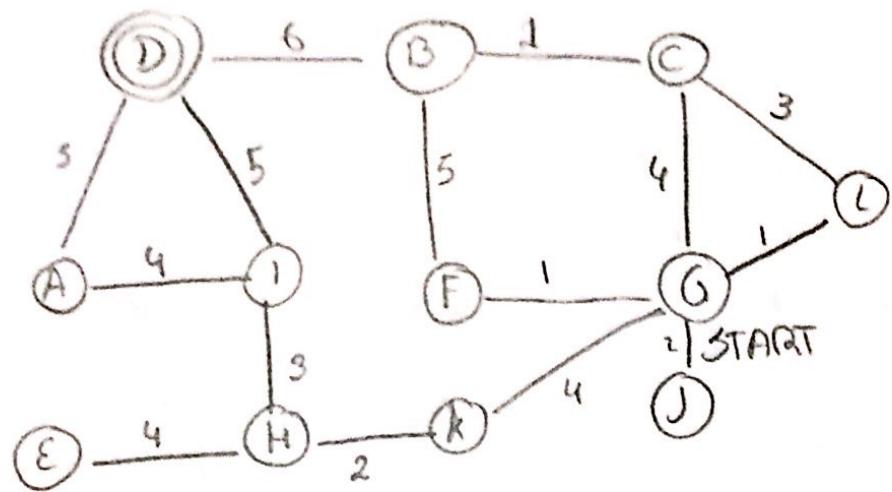
Efeito Negativo: Foco(x)

Levar(x):

Pré-condição: Tem(x)  $\wedge$  Melio Campo

Efeito Negativo:

Efeito Positivo: Melio Campo



Ordem Pesquisa: G-C-F-J-K-J-B-C-B-L-B-H-D-D-C-D-E-I  
L-A-D-

Solução: G-C-B-D