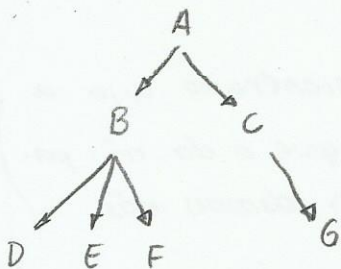


Inteligência Artificial

Buscas cegas: $\left\{ \begin{array}{l} \text{Busca em largura} \\ \text{Busca em Profundidade} \end{array} \right.$

Busca em largura (Breadth First Search) \rightarrow BFS

Fila $\{A\} \rightarrow \{B, C\} \rightarrow \{D, E, F\} \rightarrow \{D, E, F, G\}$



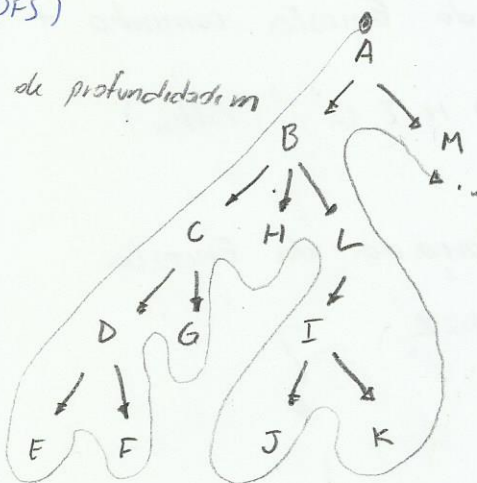
CARACTERÍSTICAS:

FIFO

- * É completa (Quando existir solução)
- * É ótima (Acha menor caminho)
- * Custo Exponencial $O(b^d)$
 - b : fator de ramificação
 - d : nível

Busca em Profundidade (Depth First Search - DFS)

- * Não é completa: Precisa definir limite de profundidade
- * Custo Exponencial $O(b^m)$

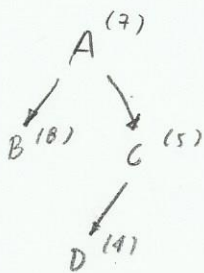


$\{A\} \rightarrow \{B, M\} \rightarrow \{C, H, L, M\} \rightarrow$
 $\{D, G, H, L, M\} \rightarrow \{E, F, G, H, L, M\} \rightarrow$
 $\{E, G, H, L, M\} \rightarrow \{G, H, L, M\} \rightarrow$
 $\{H, L, M\} \rightarrow \{L, M\} \rightarrow \{I, L, M\} \rightarrow$
 $\{J, K, L, M\} \rightarrow \dots$

Buscas Heurísticas: $\left\{ \begin{array}{l} \text{Best-First Search (Melhor-Escolha)} \\ \text{Hill-Climbing (Subida da Colina)} \\ \text{Simulated Annealing} \\ \text{A* and others.} \end{array} \right.$

Busca Melhor-escolha $\left\{ \begin{array}{l} \text{Busca Gulosa (Greedy search)} \\ \text{Algoritmo A*} \end{array} \right.$

SUBIDA DA ENCOSTA



Não é completa
 " " Ótima

Custo: $O(d) \rightarrow O(b^d)$

- * Pode Acometer 2 problemas principais:
 - Máximos locais
 - Planícies

ACD

- Exercício Slide (Aula 02) {
- Melhor Escolha
- Subida da Encosta (Normal, + Ingerme e Forçada)

(a) Melhor Escolha

A B C D H I O E F K L Q R

(Algoritmo deve abrir os nós e escolher o melhor heurística)

(b) Subida da Encosta

A B E F K L

(Expande o primeiro nó encontrado cuja a função heurística é melhor que a do nó-pai que o gerou. Ignora os demais nós)

(Porque falha?)

(c) Subida da Encosta caminho + ingerme

A B C D H I O (falha)

(Expande o melhor nó cuja a função heurística é melhor que a do nó-pai)

(d) Subida Forçada da Encosta

A B E F K L P Q R

?
0

Robotics: Computational Motion Planning

Computational Motion Planning

(CJ Taylor)

Video 1.1

The goal is to develop techniques that would allow a robot or robots to automatically decide how to move from one position or configuration to another.

Video 1.2 *Grassfire / Breadth First Search*

↳ Algorithm to get the shortest distance between the points.

For this algorithm { Start Node
Destination Node

Computational Complexity: $O(|V|)$

V : denotes the number of nodes in the graph

Examples: 2D grid $100 \times 100 \rightarrow 10^4$
3D grid $100 \times 100 \times 100 \rightarrow 10^6$

Video 1.3 *Dijkstra's Algorithm*

Also it is an algorithm for finding the shortest paths between nodes in a graph.

Computational complexity: $O(|V|^2)$

This can be reduced to $O((|E| + |V|) \log(|V|))$

V number of nodes
 E number of edges

Grassfire and Dijkstra's Algorithm visits all the nodes until they encounter the goal.

Video 1.4 *A* Search*

A* Search attempts to improve upon the performance of grassfire and Dijkstra by incorporating a heuristic function that guides the path planner.

- Heuristic functions are used to map every node in the graph to a non-negative value.
 $H(n) \leq$ length of shortest path from n to goal 01

Heuristic functions more used:

- Euclidean Distance
- Manhattan Distance

QUIZ: 1- Yes 3- Quadratically
2- No 4- A*

Assignment 1

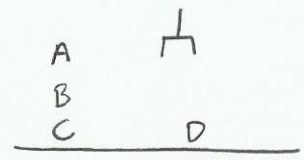
- Implement planning system
- Input: array with $\left\{ \begin{array}{l} \text{false or zero} \rightarrow \text{free cells} \\ \text{true or non-zero} \rightarrow \text{correspond to obstacle cells} \end{array} \right.$

1 2 3 4 5 6 7 8 9

1 4

1 2 3 0 1 2 3 0

Estado So



$on(a,b)$
 $on(b,c)$
 $ontable(c)$
 $clear(a)$
 $clear(D)$
 $handempty$

Ação:

desempilha (x,y)

precondição: $on(x,y) \wedge clear(x) \wedge handempty$
 remoção: $on(x,y) \wedge clear(x) \wedge handempty$
 adição: $holding(x) \wedge clear(y)$

desempilha (a,b)

<u>Pre</u>	<u>Remoção</u>	<u>Adição</u>
$on(a,b) \times$	$on(b,c)$	$on(b,c)$
$on(b,c)$	$ontable(c)$	$ontable(c)$
$ontable(c)$	$clear(D)$	$clear(D)$
$clear(a) \times$		$holding(a)$
$clear(D)$		$clear(b)$
$handempty \times$		

Empilhar (x,y)

Pre: $clear(y) \wedge holding(x)$
 Remoção: $clear(y) \wedge holding(x)$
 Adição: $handempty \wedge on(x,y)$

Pegar (x)

Pre: $handempty \wedge clear(x)$
 Remoção: " "
 Adição: $holding(x)$

Soltar (x)

Pre: $holding(x)$
 Remoção: "
 Adição: $handempty \wedge ontable(x)$

Inteligência Artificial

Carlos Eduardo Thomaz
cet@fei.edu.br
http://www.fei.edu.br/~cet

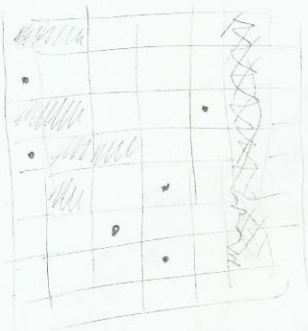
- Russel Stuart J; Norvig Peter. Inteligência Artificial. 2004

- Visual Turing Test. → new scientist

- Sokoban
- Sudoku
- Pac man

MECANISMO DE BUSCA

24/02/2016



Tela 8xc

Busca: encontrar solução de um problema através de um espaço de soluções.

- A busca é feita em uma árvore (grafo) de busca
- Espaço de busca é o conjunto de soluções
- Estado é uma descrição de uma situação (solução)

5	4	
6	1	8
7	3	2

Grafo: bidirecional

Árvore: unidirecional (Dificuldade para loops)

Vetor: [5, 4, *, 6, 1, 8, 7, 3, 2]

" 12 13 21 22 23 31 32 33

Heurística: Informação, valor e/ou quantificação de um estado

quanto sua relevância ou distância de outros estados

Conceito popular: "chute Inicial"

→ Evita explosão combinatoria

(ver Planices (platóis))

Next Class: Algoritmo A*

Simulated Annealing

Solução para quando ocorrer falha ao encontrar um máximo-local

$$p(x) = \alpha e^{E(x)/kT}$$

Heurística Admissível

$$h(n) \leq h^*(n)$$

Heurística (estimativa) Custo real

(Manter valor para detalhes mais aprofundados)

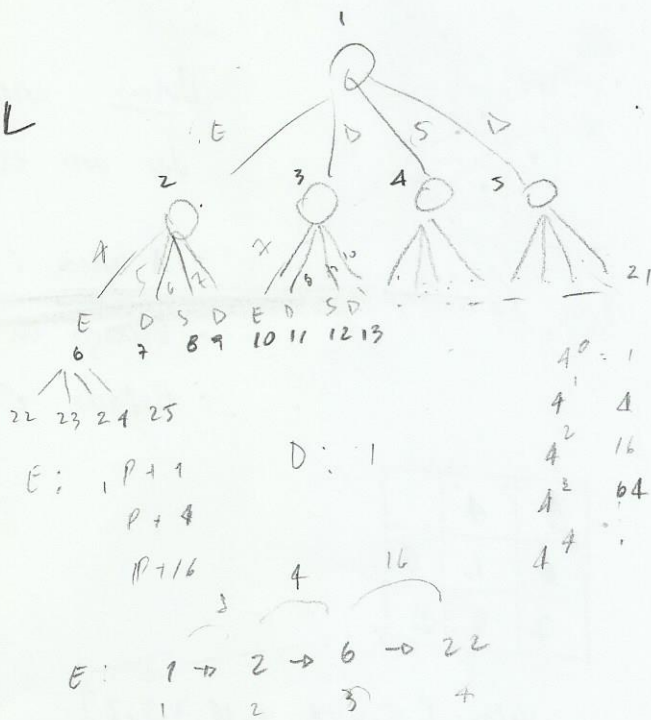
Exercício

A D I E L

- {A}
- {B, C, D}
- {D, B, C}
- {H, E, B, C}
- ordem
- ⋮

Estratégia MAX/MIN

A e B



Data Mining e Aprendizado

O Data Mining é uma ferramenta para extração de conhecimento a partir de grandes bases de dados. (KDD)

Estrutura de um KDD com Data Mining

1 Dados 2 Seleção
3 Processamento 4 Transformação 5 Mineração de Dados 6 Interpretação e Avaliação Conhecimento

Data Mining (DM) visa extrair conhecimento (padrões, comportamentos, relações, etc) de uma Data Warehouse.

Exemplos de técnicas utilizadas:

- Redes neurais
- Indução de Regras
- Árvores de Decisão
- Análise de Cluster
- Análise de Séries Temporais
- Visualização
- Técnicas de algoritmos genéticos

Exercício 1)

(a) Mecanismo de Busca em largura e Profundidade

Largura (BFS):

- Completa: Sempre acha uma solução se existir.
- É ótima
- Custo $O(b^d)$ Exponencial

Profundidade

- NÃO é completa (Pecusa de um limite de profundidade)
- NÃO é ótima
- Custo $O(b^d)$ - (Pior caso)

- (b) Buscas heurísticas
- Best-First Search (Melhor Escolha)
 - Hill-Climbing (Subida da Colina)
 - Simulated Annealing
 - A*
 - ...

Mecanismos de busca baseados em heurísticas não são sempre ótimos

Exemplo: "Melhor escolha" Não garante que seja ótimo.

(c) Aprendizado supervisionado:

"Dado um conjunto de exemplos pré-classificados, aprender uma descrição geral que encapsula a informação contida nesses exemplos e que pode ser usada para prever casos futuros"

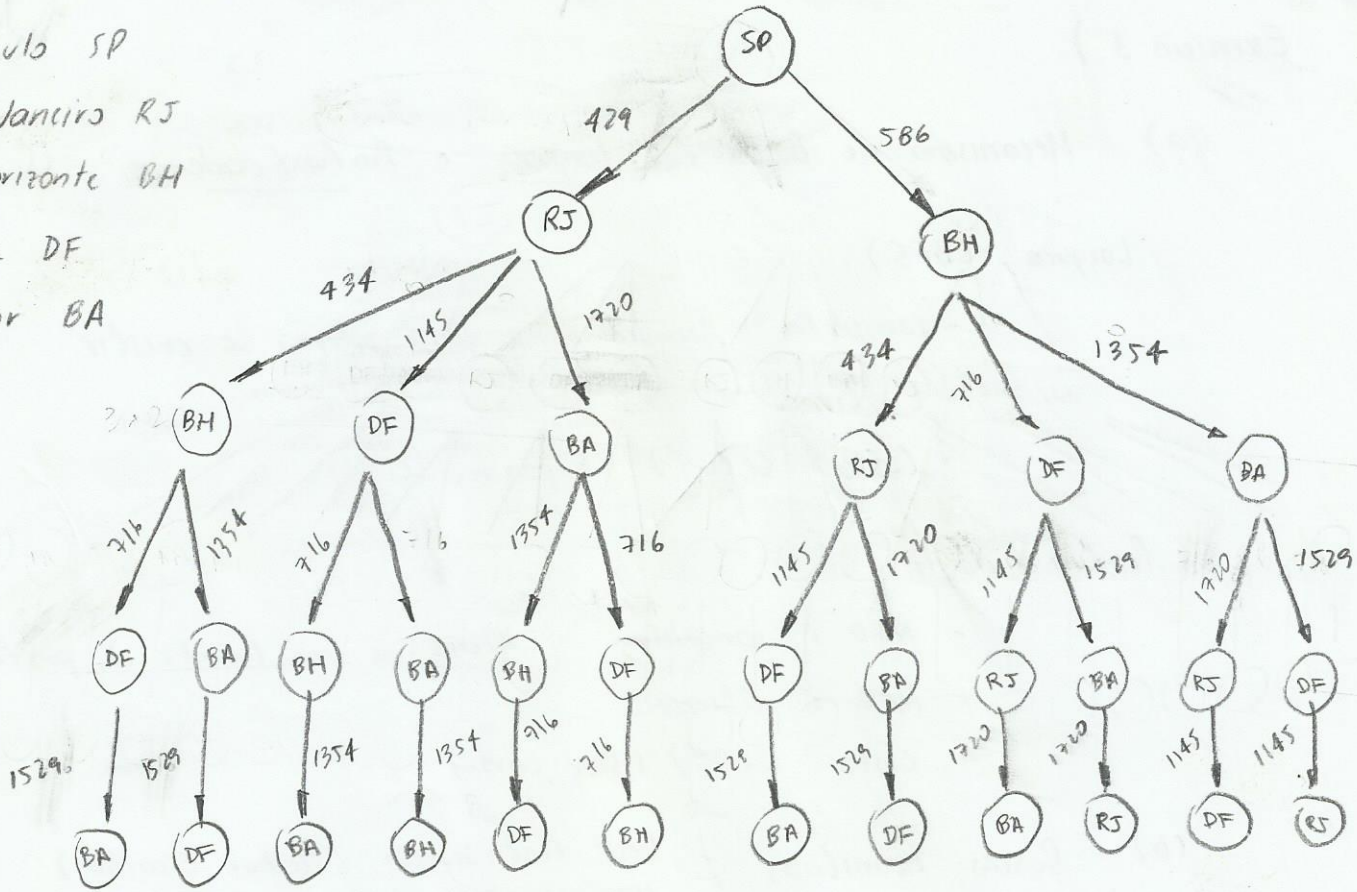
O paradigma do aprendizado indutivo é o aprendizado por reforço

RESPOSTA:

(a) e (b)

Exercício 2)

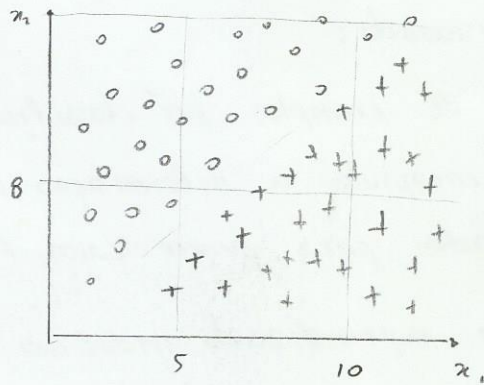
São Paulo SP
 Rio de Janeiro RJ
 Belo Horizonte BH
 Brasília DF
 Salvador BA



Sequência (Melhor Escolha)

SP → RJ → BH → DF → BA

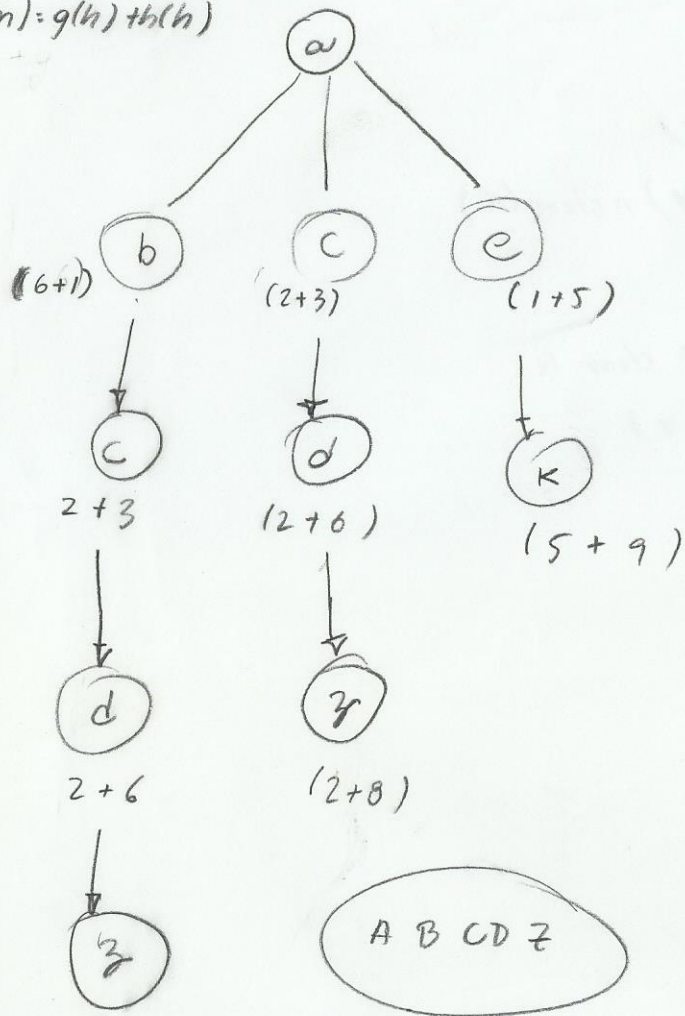
Exercício 3)



Ponto (6,7) é 'o' ou '+'

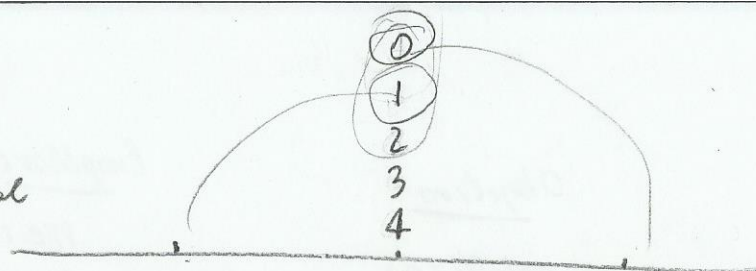
Questão 5

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

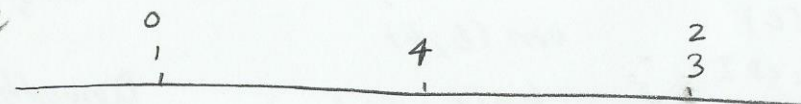


Questão 6

Inicial



Final



Estado Inicial

- entro (4)
- on (3,4)
- on (2,3) x
- on (1,2) x
- on (0,1) x

S₁

esquerda (0) x

S₂

direita (1)

S₃

on (0,1)

S₄

esquerda (2)

S₅

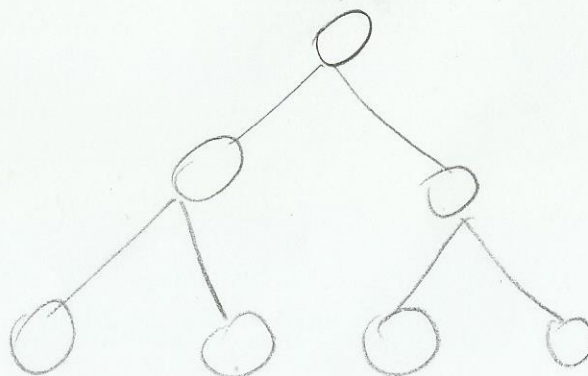
on (0,1)

⑥ on (2,3)

⑦ direita (0)

⑧ on (1,2)

[43210]



Questão 7

Início

on(B,A)
ontable(A)
ontable(C)
ontable(D)
handempty
clear(B)
clear(C)
clear(D)

Objetivo

ontable(A)
ontable(D)
on(B,D)
holding(C)

Empilhador(x,y)

PRE: $\text{holding}(x) \wedge \text{clear}(y)$

Remoção: $\text{holding}(x) \wedge \text{clear}(y)$

Adição: $\text{handempty} \wedge \text{on}(x,y) \wedge \text{clear}(x)$

Desempilhador(x,y)

PRE: $\text{handempty} \wedge \text{on}(x,y) \wedge \text{clear}(x)$

REMOÇÃO: $\text{handempty} \wedge \text{on}(x,y)$

ADICÃO: $\text{clear}(x)$

Início

S.

on(B,A)
ontable(A)
" (C)
" (D)
handempty
clear(B)
" (C)
" (D)