Inteligência Artificial
Aula 5 - vídeo 3 - Buscas Não Informadas

João C. P. da Silva

Dept. Ciência da Computação - UFRJ

11 de setembro de 2020

Algoritmo de Busca Genérico

Algoritmo de Busca Genérico

- dado um grafo, nós iniciais e nós finais, explore caminhos a partir dos nós iniciais.

Algoritmo de Busca Genérico

- dado um grafo, nós iniciais e nós finais, explore caminhos a partir dos nós iniciais.
- Mantenha uma fronteira dos caminhos a partir do nó inicial que já foram explorados.

Algoritmo de Busca Genérico

- dado um grafo, nós iniciais e nós finais, explore caminhos a partir dos nós iniciais.
- Mantenha uma fronteira dos caminhos a partir do nó inicial que já foram explorados.
- Conforme a busca prossegue, a fronteira avança sobre os nós não explorados, até que um nó objetivo é encontrado.

Algoritmo de Busca Genérico

- dado um grafo, nós iniciais e nós finais, explore caminhos a partir dos nós iniciais.
- Mantenha uma fronteira dos caminhos a partir do nó inicial que já foram explorados.
- Conforme a busca prossegue, a fronteira avança sobre os nós não explorados, até que um nó objetivo é encontrado.
- A maneira como a fronteira é expandida define a estratégia de busca.

Algoritmo de Busca Genérico

- dado um grafo, nós iniciais e nós finais, explore caminhos a partir dos nós iniciais.
- Mantenha uma fronteira dos caminhos a partir do nó inicial que já foram explorados.
- Conforme a busca prossegue, a fronteira avança sobre os nós não explorados, até que um nó objetivo é encontrado.
- A maneira como a fronteira é expandida define a estratégia de busca.

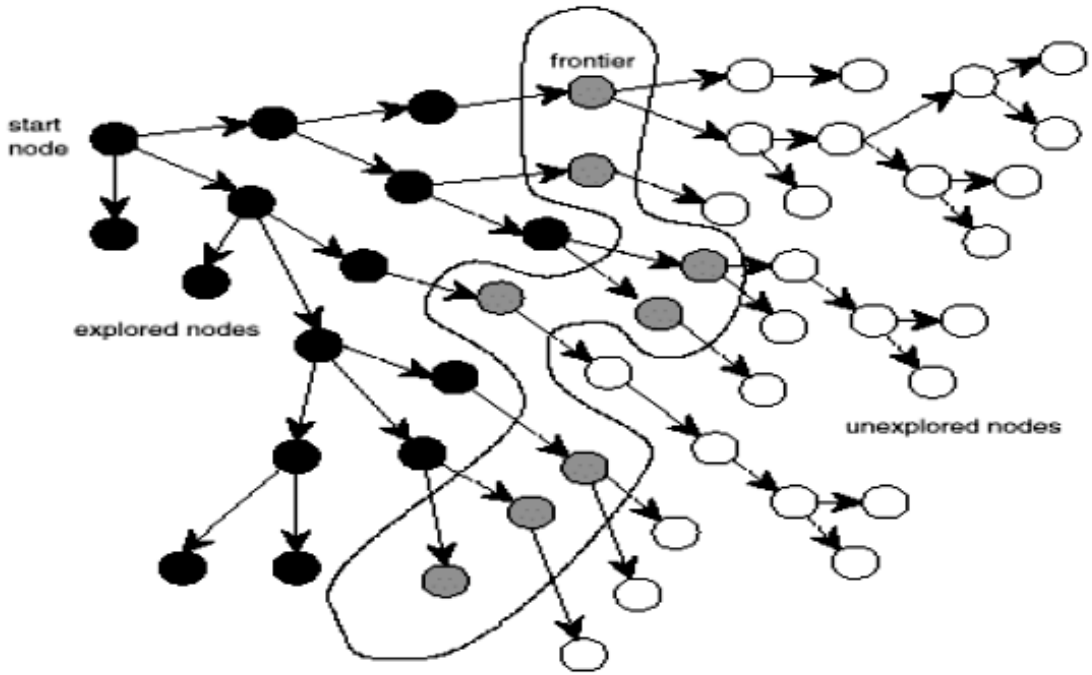


Figura: Fonte: Computational Intelligence: A Logical Approach, Poole, Mackworth and Goebel

Buscas Não-Informadas

Utilizam somente informações disponíveis na definição do problema.

Estratégias

- Largura
- Profundidade
- Profundidade-Limitada
- Profundidade Iterativa
- Custo-Uniforme
- Bidirecional

Busca em Largura

Tratar a fronteira como uma fila.

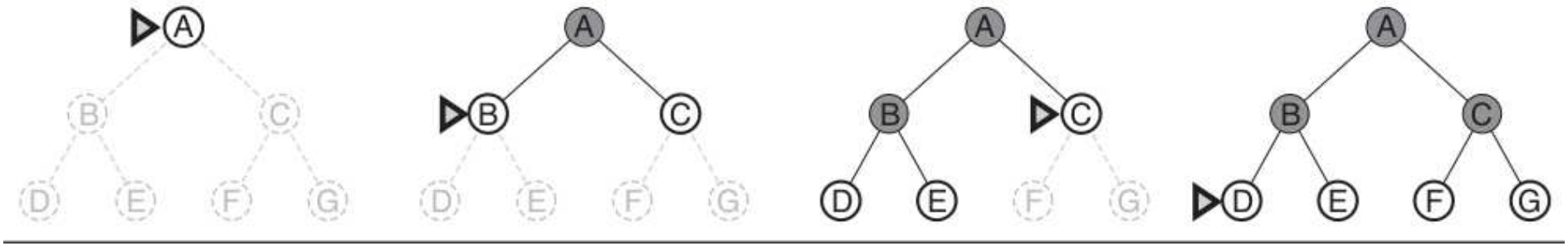


Figura: Fonte: Artificial Intelligence: A Modern Approach - Russell and Norvig

Busca em Profundidade

Tratar a fronteira como uma pilha.

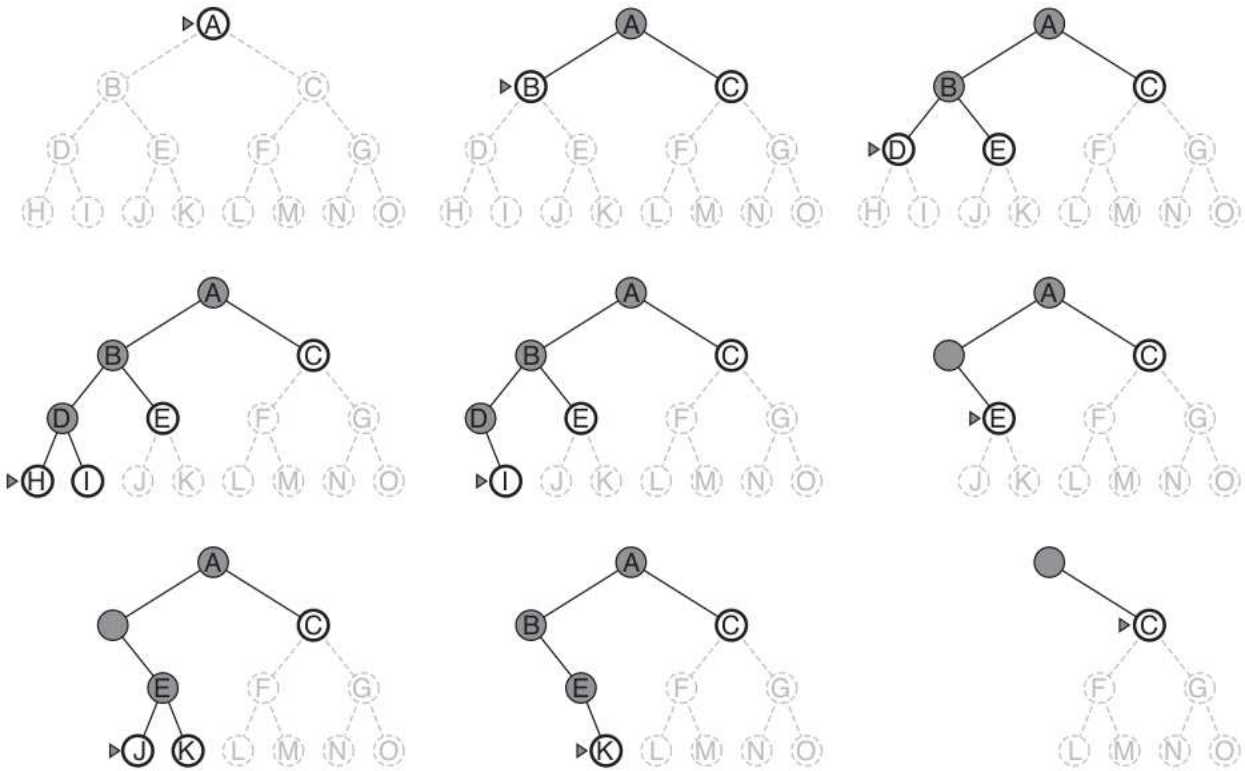


Figura: Fonte: Artificial Intelligence: A Modern Approach - Russell and Norvig

Problema das Jarras - Busca em Largura

(0,0)

Problema das Jarras - Busca em Largura

(0,0)

Regras

R_1: $T \Rightarrow J_1$

R_2: $T \Rightarrow J_2$

R_3: $J_1 \Rightarrow F$

R_4: $J_2 \Rightarrow F$

R_5: $J_1 \Rightarrow J_2$

R_6: $J_2 \Rightarrow J_1$

Problema das Jarras - Busca em Largura

(0,0)

Fronteira

[(0,0)]

Regras

R_1: $T \Rightarrow J_1$

R_2: $T \Rightarrow J_2$

R_3: $J_1 \Rightarrow F$

R_4: $J_2 \Rightarrow F$

R_5: $J_1 \Rightarrow J_2$

R_6: $J_2 \Rightarrow J_1$

Problema das Jarras - Busca em Largura

(0,0)

Fronteira

[(0,0)]

Regras

R_1: T ⇒ J_1

R_2: T ⇒ J_2

R_3: J_1 ⇒ F

R_4: J_2 ⇒ F

R_5: J_1 ⇒ J_2

R_6: J_2 ⇒ J_1

Problema das Jarras - Busca em Largura

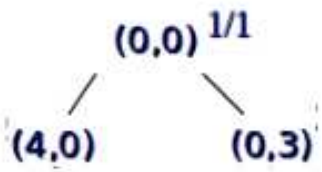
Ordem Geração / Ordem Expansão	Fronteira	Regras
(0,0)	[(0,0)]	R_1: T ⇒ J_1
		R_2: T ⇒ J_2
		R_3: J_1 ⇒ F
		R_4: J_2 ⇒ F
		R_5: J_1 ⇒ J_2
		R_6: J_2 ⇒ J_1

Problema das Jarras - Busca em Largura

Ordem Geração / Ordem Expansão	Fronteira	Regras
$(0,0)^{1/1}$	$[\underline{(0,0)}]$	$R_1: T \Rightarrow J_1$
		$R_2: T \Rightarrow J_2$
		$R_3: J_1 \Rightarrow F$
		$R_4: J_2 \Rightarrow F$
		$R_5: J_1 \Rightarrow J_2$
		$R_6: J_2 \Rightarrow J_1$

Problema das Jarras - Busca em Largura

Ordem Geração / Ordem Expansão



Fronteira

[(0,0)]

Regras

R_1: T ⇒ J_1

R_2: T ⇒ J_2

R_3: J_1 ⇒ F

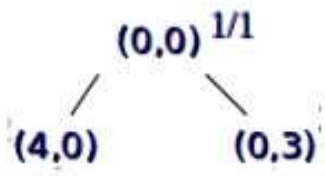
R_4: J_2 ⇒ F

R_5: J_1 ⇒ J_2

R_6: J_2 ⇒ J_1

Problema das Jarras - Busca em Largura

Ordem Geração / Ordem Expansão



Fronteira

[(0,0)]

[(4,0), (0,3)]

Regras

R_1: $T \Rightarrow J_1$

R_2: $T \Rightarrow J_2$

R_3: $J_1 \Rightarrow F$

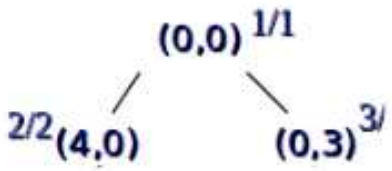
R_4: $J_2 \Rightarrow F$

R_5: $J_1 \Rightarrow J_2$

R_6: $J_2 \Rightarrow J_1$

Problema das Jarras - Busca em Largura

Ordem Geração / Ordem Expansão



Fronteira

[(0,0)]

[(4,0), (0,3)]

Regras

R_1: T ⇒ J_1

R_2: T ⇒ J_2

R_3: J_1 ⇒ F

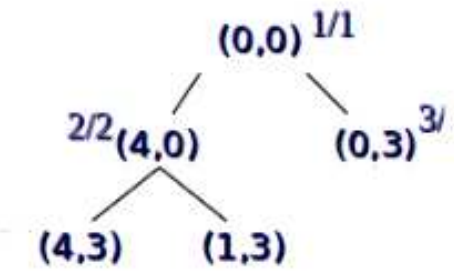
R_4: J_2 ⇒ F

R_5: J_1 ⇒ J_2

R_6: J_2 ⇒ J_1

Problema das Jarras - Busca em Largura

Ordem Geração / Ordem Expansão



Fronteira

[(0,0)]

[(4,0), (0,3)]

Regras

R_1: T ⇒ J_1

R_2: T ⇒ J_2

R_3: J_1 ⇒ F

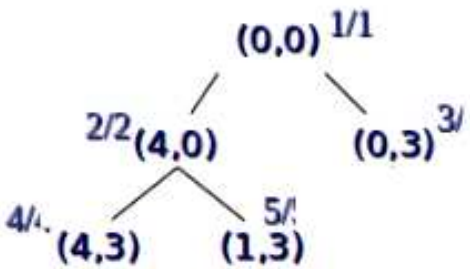
R_4: J_2 ⇒ F

R_5: J_1 ⇒ J_2

R_6: J_2 ⇒ J_1

Problema das Jarras - Busca em Largura

Ordem Geração / Ordem Expansão



Fronteira

[(0,0)]

[(4,0), (0,3)]

Regras

R_1: T $\Rightarrow$ J_1

R_2: T $\Rightarrow$ J_2

R_3: J_1 $\Rightarrow$ F

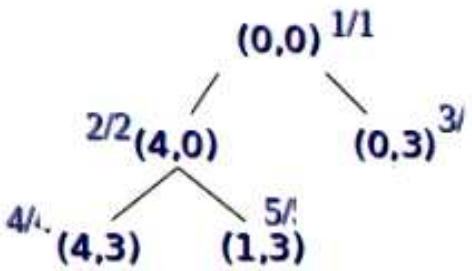
R_4: J_2 $\Rightarrow$ F

R_5: J_1 $\Rightarrow$ J_2

R_6: J_2 $\Rightarrow$ J_1

Problema das Jarras - Busca em Largura

Ordem Geração / Ordem Expansão



Fronteira

[(0,0)]

[(4,0), (0,3)]

[(0,3), (4,3), (1,3)]

Regras

R_1: T ⇒ J_1

R_2: T ⇒ J_2

R_3: J_1 ⇒ F

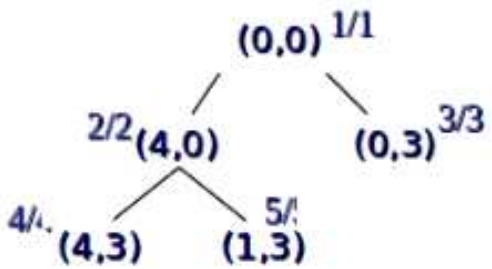
R_4: J_2 ⇒ F

R_5: J_1 ⇒ J_2

R_6: J_2 ⇒ J_1

Problema das Jarras - Busca em Largura

Ordem Geração / Ordem Expansão



Fronteira

[(0,0)]

[(4,0), (0,3)]

[(0,3), (4,3), (1,3)]

Regras

R_1: T ⇒ J_1

R_2: T ⇒ J_2

R_3: J_1 ⇒ F

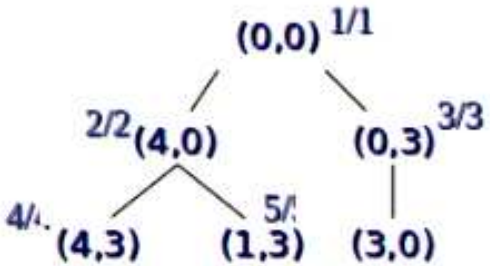
R_4: J_2 ⇒ F

R_5: J_1 ⇒ J_2

R_6: J_2 ⇒ J_1

Problema das Jarras - Busca em Largura

Ordem Geração / Ordem Expansão



Fronteira

[(0,0)]

[(4,0), (0,3)]

[(0,3), (4,3), (1,3)]

Regras

R_1: T $\Rightarrow J_1$

R_2: T $\Rightarrow J_2$

R_3: $J_1 \Rightarrow$ F

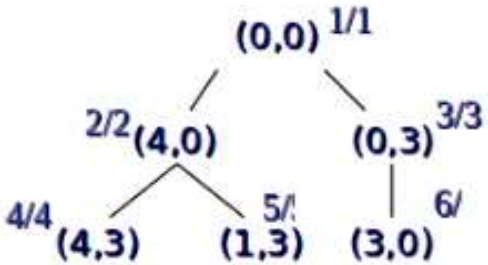
R_4: $J_2 \Rightarrow$ F

R_5: $J_1 \Rightarrow J_2$

R_6: $J_2 \Rightarrow J_1$

Problema das Jarras - Busca em Largura

Ordem Geração / Ordem Expansão



Fronteira

[(0,0)]

[(4,0), (0,3)]

[(0,3), (4,3), (1,3)]

[(4,3), (1,3), (3,0)]

Regras

R_1: T ⇒ J_1

R_2: T ⇒ J_2

R_3: J_1 ⇒ F

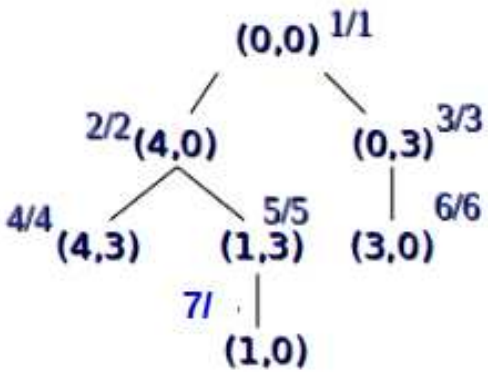
R_4: J_2 ⇒ F

R_5: J_1 ⇒ J_2

R_6: J_2 ⇒ J_1

Problema das Jarras - Busca em Largura

Ordem Geração / Ordem Expansão



Fronteira

[(0,0)]

[(4,0), (0,3)]

[(0,3), (4,3), (1,3)]

[(4,3), (1,3), (3,0)]

[(1,3), (3,0)]

[(3,0), (1,0)]

Regras

R_1: T $\Rightarrow J_1$

R_2: T $\Rightarrow J_2$

R_3: $J_1 \Rightarrow$ F

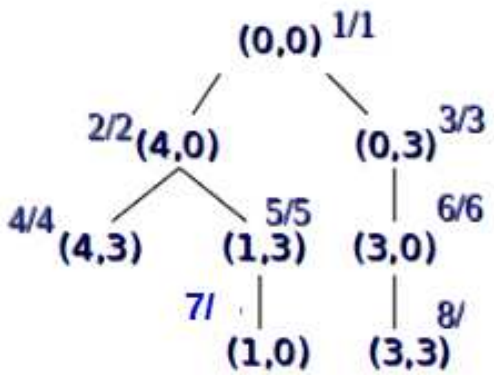
R_4: $J_2 \Rightarrow$ F

R_5: $J_1 \Rightarrow J_2$

R_6: $J_2 \Rightarrow J_1$

Problema das Jarras - Busca em Largura

Ordem Geração / Ordem Expansão



Fronteira

[(0,0)]

[(4,0), (0,3)]

[(0,3), (4,3), (1,3)]

[(4,3), (1,3), (3,0)]

[(1,3), (3,0)]

[(3,0), (1,0)]

[(1,0), (3,3)]

Regras

R_1: T $\Rightarrow$ J_1

R_2: T $\Rightarrow$ J_2

R_3: J_1 $\Rightarrow$ F

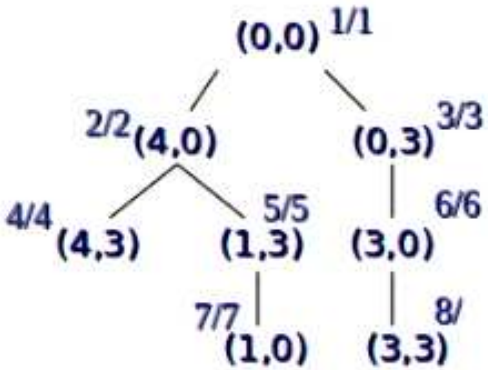
R_4: J_2 $\Rightarrow$ F

R_5: J_1 $\Rightarrow$ J_2

R_6: J_2 $\Rightarrow$ J_1

Problema das Jarras - Busca em Largura

Ordem Geração / Ordem Expansão



Fronteira

[(0,0)]

[(4,0), (0,3)]

[(0,3), (4,3), (1,3)]

[(4,3), (1,3), (3,0)]

[(1,3), (3,0)]

[(3,0), (1,0)]

[(1,0), (3,3)]

Regras

R_1: $T \Rightarrow J_1$

R_2: $T \Rightarrow J_2$

R_3: $J_1 \Rightarrow F$

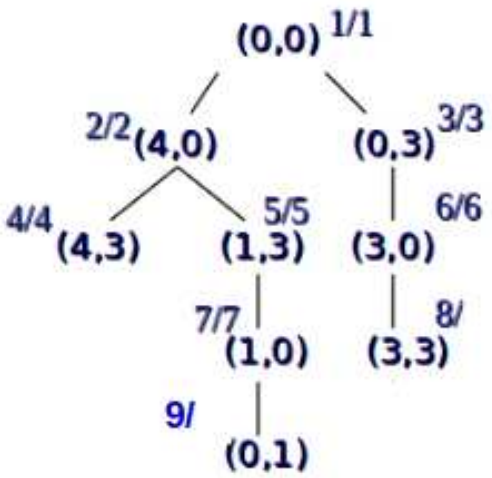
R_4: $J_2 \Rightarrow F$

R_5: $J_1 \Rightarrow J_2$

R_6: $J_2 \Rightarrow J_1$

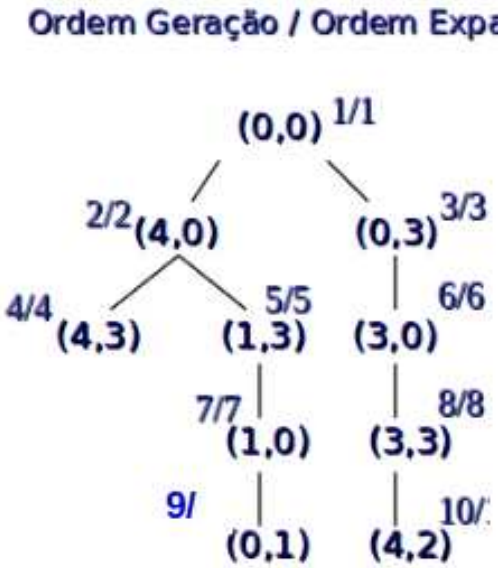
Problema das Jarras - Busca em Largura

Ordem Geração / Ordem Expansão



Fronteira	Regras
[(0,0)]	R_1: T ⇒ J_1
[(4,0), (0,3)]	R_2: T ⇒ J_2
[(0,3), (4,3), (1,3)]	R_3: J_1 ⇒ F
[(4,3), (1,3), (3,0)]	R_4: J_2 ⇒ F
[(1,3), (3,0)]	R_5: J_1 ⇒ J_2
[(3,0), (1,0)]	R_6: J_2 ⇒ J_1
[(1,0), (3,3)]	
[(3,3), (0,1)]	

Problema das Jarras - Busca em Largura



Fronteira

[(0,0)]

[(4,0), (0,3)]

[(0,3), (4,3), (1,3)]

[(4,3), (1,3), (3,0)]

[(1,3), (3,0)]

[(3,0), (1,0)]

[(1,0), (3,3)]

[(3,3), (0,1)]

Regras

R_1: T $\Rightarrow$ J_1

R_2: T $\Rightarrow$ J_2

R_3: J_1 $\Rightarrow$ F

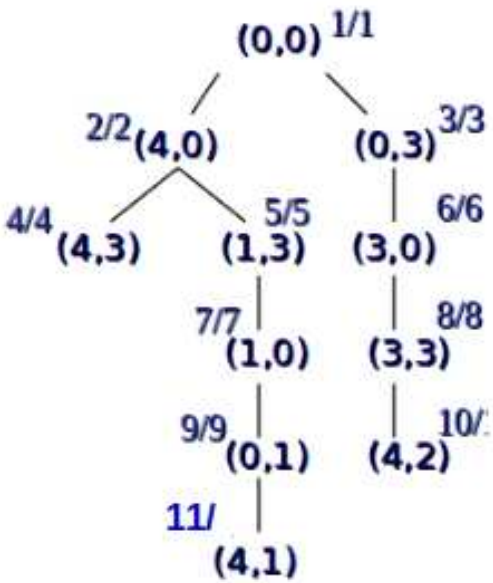
R_4: J_2 $\Rightarrow$ F

R_5: J_1 $\Rightarrow$ J_2

R_6: J_2 $\Rightarrow$ J_1

Problema das Jarras - Busca em Largura

Ordem Geração / Ordem Expansão



Fronteira

[(0,0)]

[(4,0), (0,3)]

[(0,3), (4,3), (1,3)]

[(4,3), (1,3), (3,0)]

[(1,3), (3,0)]

[(3,0), (1,0)]

[(1,0), (3,3)]

[(3,3), (0,1)]

[(0,1), (4,2)]

[(4,2), (4,1)]

Regras

R_1: T $\Rightarrow$ J_1

R_2: T $\Rightarrow$ J_2

R_3: J_1 $\Rightarrow$ F

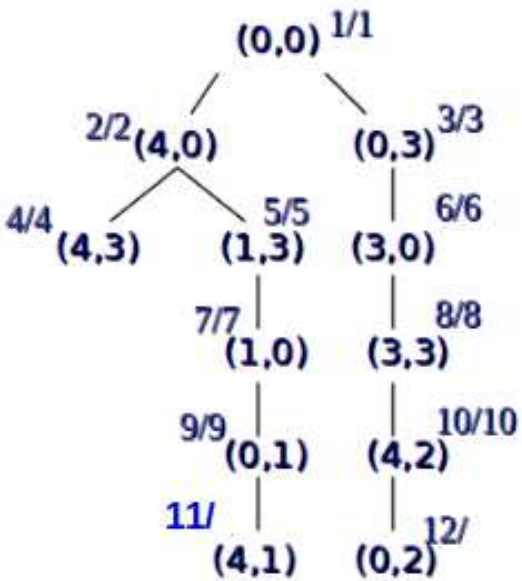
R_4: J_2 $\Rightarrow$ F

R_5: J_1 $\Rightarrow$ J_2

R_6: J_2 $\Rightarrow$ J_1

Problema das Jarras - Busca em Largura

Ordem Geração / Ordem Expansão



Fronteira

[(0,0)]

[(4,0), (0,3)]

[(0,3), (4,3), (1,3)]

[(4,3), (1,3), (3,0)]

[(1,3), (3,0)]

[(3,0), (1,0)]

[(1,0), (3,3)]

[(3,3), (0,1)]

[(0,1), (4,2)]

[(4,2), (4,1)]

Regras

R_1: T $\Rightarrow$ J_1

R_2: T $\Rightarrow$ J_2

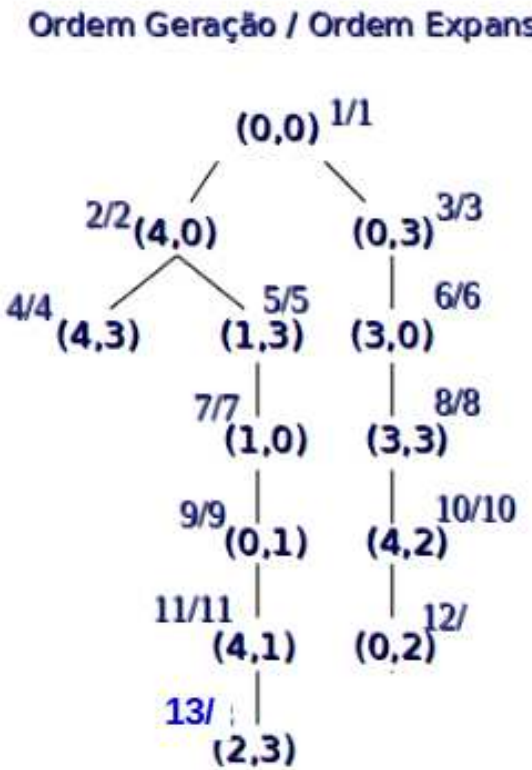
R_3: J_1 $\Rightarrow$ F

R_4: J_2 $\Rightarrow$ F

R_5: J_1 $\Rightarrow$ J_2

R_6: J_2 $\Rightarrow$ J_1

Problema das Jarras - Busca em Largura



Fronteira

[(0,0)]

[(4,0), (0,3)]

[(0,3), (4,3), (1,3)]

[(4,3), (1,3), (3,0)]

[(1,3), (3,0)]

[(3,0), (1,0)]

[(1,0), (3,3)]

[(3,3), (0,1)]

[(0,1), (4,2)]

[(4,2), (4,1)]

[(4,1), (0,2)]

Regras

R_1: T $\Rightarrow J_1$

R_2: T $\Rightarrow J_2$

R_3: $J_1 \Rightarrow$ F

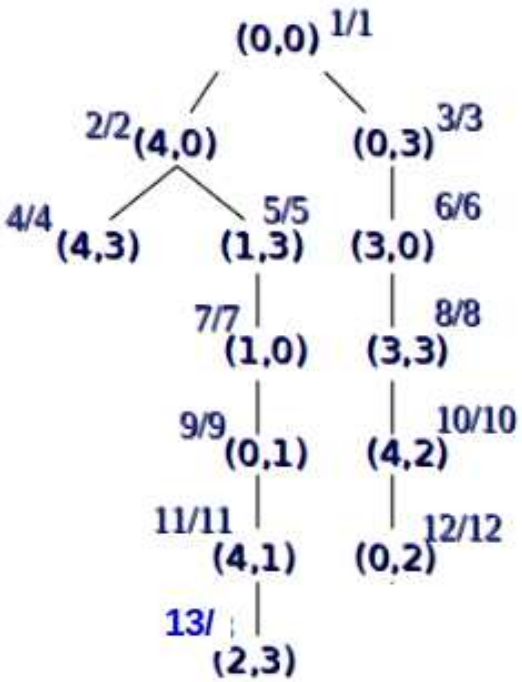
R_4: $J_2 \Rightarrow$ F

R_5: $J_1 \Rightarrow J_2$

R_6: $J_2 \Rightarrow J_1$

Problema das Jarras - Busca em Largura

Ordem Geração / Ordem Expansão



Fronteira

[(0,0)]

[(4,0), (0,3)]

[(0,3), (4,3), (1,3)]

[(4,3), (1,3), (3,0)]

[(1,3), (3,0)]

[(3,0), (1,0)]

[(1,0), (3,3)]

[(3,3), (0,1)]

[(0,1), (4,2)]

[(4,2), (4,1)]

[(4,1), (0,2)]

[(0,2), (2,3)]

Regras

R_1: T $\Rightarrow$ J_1

R_2: T $\Rightarrow$ J_2

R_3: $J_1 \Rightarrow$ F

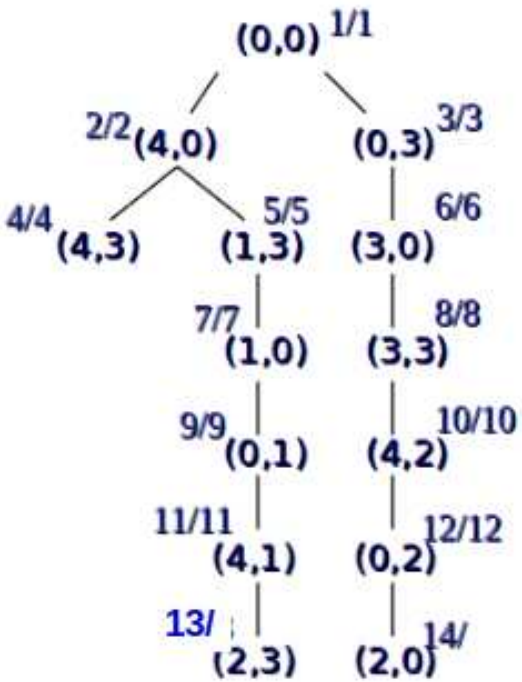
R_4: $J_2 \Rightarrow$ F

R_5: $J_1 \Rightarrow J_2$

R_6: $J_2 \Rightarrow J_1$

Problema das Jarras - Busca em Largura

Ordem Geração / Ordem Expansão



Fronteira

[(0,0)]

[(4,0), (0,3)]

[(0,3), (4,3), (1,3)]

[(4,3), (1,3), (3,0)]

[(1,3), (3,0)]

[(3,0), (1,0)]

[(1,0), (3,3)]

[(3,3), (0,1)]

[(0,1), (4,2)]

[(4,2), (4,1)]

[(4,1), (0,2)]

[(0,2), (2,3)]

Regras

R_1: T $\Rightarrow$ J_1

R_2: T $\Rightarrow$ J_2

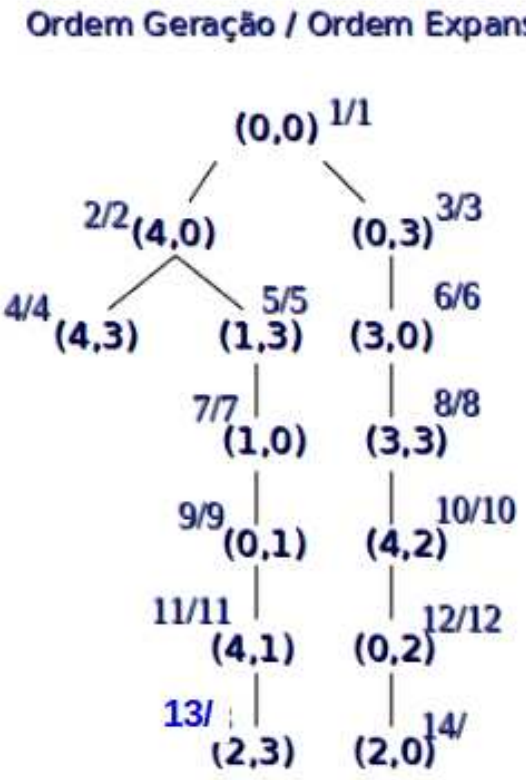
R_3: J_1 $\Rightarrow$ F

R_4: J_2 $\Rightarrow$ F

R_5: J_1 $\Rightarrow$ J_2

R_6: J_2 $\Rightarrow$ J_1

Problema das Jarras - Busca em Largura



Fronteira

[(0,0)]

[(4,0), (0,3)]

[(0,3), (4,3), (1,3)]

[(4,3), (1,3), (3,0)]

[(1,3), (3,0)]

[(3,0), (1,0)]

[(1,0), (3,3)]

[(3,3), (0,1)]

[(0,1), (4,2)]

[(4,2), (4,1)]

[(4,1), (0,2)]

[(0,2), (2,3)]

[(2,3), (2,0)]

Regras

R_1: T $\Rightarrow$ J_1

R_2: T $\Rightarrow$ J_2

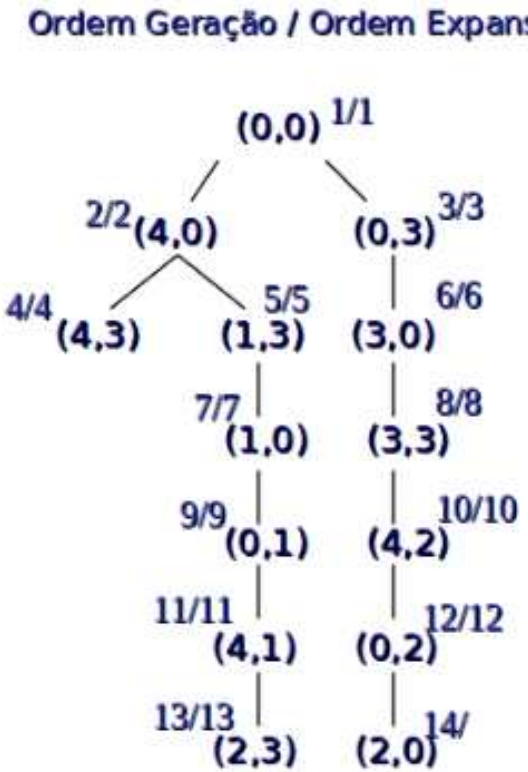
R_3: J_1 $\Rightarrow$ F

R_4: J_2 $\Rightarrow$ F

R_5: J_1 $\Rightarrow$ J_2

R_6: J_2 $\Rightarrow$ J_1

Problema das Jarras - Busca em Largura



Fronteira

[(0,0)]

[(4,0), (0,3)]

[(0,3), (4,3), (1,3)]

[(4,3), (1,3), (3,0)]

[(1,3), (3,0)]

[(3,0), (1,0)]

[(1,0), (3,3)]

[(3,3), (0,1)]

[(0,1), (4,2)]

[(4,2), (4,1)]

[(4,1), (0,2)]

[(0,2), (2,3)]

[(2,3), (2,0)]

Regras

R_1: T $\Rightarrow$ J_1

R_2: T $\Rightarrow$ J_2

R_3: J_1 $\Rightarrow$ F

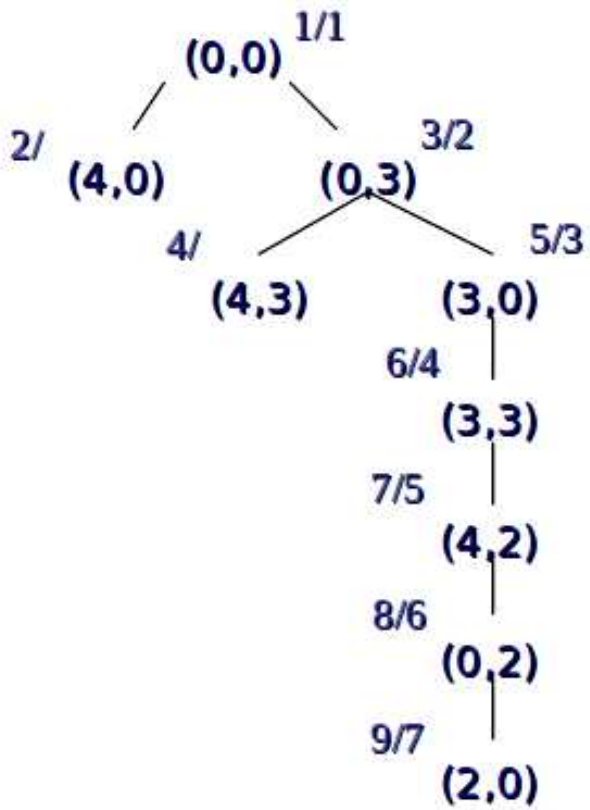
R_4: J_2 $\Rightarrow$ F

R_5: J_1 $\Rightarrow$ J_2

R_6: J_2 $\Rightarrow$ J_1

Problema das Jarras - Busca em Profundidade

Ordem Geração / Ordem Expansão



Fronteira

[(0,0)]

[(0,3), (4,0)]

[(3,0), (4,3), (4,0)]

[(3,3), (4,3), (4,0)]

[(4,2), (4,3), (4,0)]

[(0,2), (4,3), (4,0)]

[(2,0), (4,3), (4,0)]

Regras

R_1: T ⇒ J_1

R_2: T ⇒ J_2

R_3: J_1 ⇒ F

R_4: J_2 ⇒ F

R_5: J_1 ⇒ J_2

R_6: J_2 ⇒ J_1

Algoritmo de Busca Genérico

$search([Node \mid \_]) :- is\_goal(Node).$

$search([Node \mid F_1]) :- neighbors(Node, NN),$
$add\_to\_frontier(NN, F_1, F_2),$
$search(F_2).$

Algoritmo de Busca Genérico

$$search([Node \mid \_]) \text{ :- } is\_goal(Node).$$

$$search([Node \mid F_1]) \text{ :- } neighbors(Node, NN),\ add\_to\_frontier(NN, F_1, F_2),\ search(F_2).$$

- $search(F_0)$: **verdadeiro** se existe um caminho de um elemento da fronteira F_0 até um nó objetivo. A fronteira F_0 é representada por uma lista Prolog.

Algoritmo de Busca Genérico

$$search([Node \mid \_]) \text{ :- } is\_goal(Node).$$

$$search([Node \mid F_1]) \text{ :- } neighbors(Node, NN),\ add\_to\_frontier(NN, F_1, F_2),\ search(F_2).$$

- $search(F_0)$: **verdadeiro** se existe um caminho de um elemento da fronteira F_0 até um nó objetivo. A fronteira F_0 é representada por uma lista Prolog.
- $is\_goal(Node)$: **verdadeiro** se *Node* é um nó objetivo.

Algoritmo de Busca Genérico

$search([Node \mid \_]) :- is\_goal(Node).$

$search([Node \mid F_1]) :- neighbors(Node, NN),$
$\quad add\_to\_frontier(NN, F_1, F_2),$
$\quad search(F_2).$

- $search(F_0)$: **verdadeiro** se existe um caminho de um elemento da fronteira F_0 até um nó objetivo. A fronteira F_0 é representada por uma lista Prolog.
- $is\_goal(Node)$: **verdadeiro** se $Node$ é um nó objetivo.
- $neighbors(Node, NN)$: **verdadeiro** se NN é a lista de vizinhos de $Node$.

Algoritmo de Busca Genérico

$search([Node \mid \_]) :- is\_goal(Node).$

$search([Node \mid F_1]) :- neighbors(Node, NN),$
$add\_to\_frontier(NN, F_1, F_2),$
$search(F_2).$

- $search(F_0)$: **verdadeiro** se existe um caminho de um elemento da fronteira F_0 até um nó objetivo. A fronteira F_0 é representada por uma lista Prolog.
- $is\_goal(Node)$: **verdadeiro** se $Node$ é um nó objetivo.
- $neighbors(Node, NN)$: **verdadeiro** se NN é a lista de vizinhos de $Node$.
- $add\_to\_frontier(NN, F_1, F_2)$: **verdadeiro** se $F_2 = F_1 \cup NN$.

Algoritmo de Busca Genérico

$search([Node \mid \_]) :- is\_goal(Node).$

$search([Node \mid F_1]) :- neighbors(Node, NN),$
$add\_to\_frontier(NN, F_1, F_2),$
$search(F_2).$

- $search(F_0)$: **verdadeiro** se existe um caminho de um elemento da fronteira F_0 até um nó objetivo. A fronteira F_0 é representada por uma lista Prolog.
- $is\_goal(Node)$: **verdadeiro** se $Node$ é um nó objetivo.
- $neighbors(Node, NN)$: **verdadeiro** se NN é a lista de vizinhos de $Node$.
- $add\_to\_frontier(NN, F_1, F_2)$: **verdadeiro** se $F_2 = F_1 \cup NN$.

A estratégia de busca é definida dependendo de como a fronteira é construída.

Buscas em Largura e Profundidade

$search([Node \mid \_]) :- is\_goal(Node).$

$search([Node \mid F_1]) :- neighbors(Node, NN),$
$add\_to\_frontier(NN, F_1, F_2),$
$search(F_2).$

- **Largura**: a fronteira é uma fila, ou seja, os novos nós entram no final da lista.

$$add\_to\_frontier(NN, F_1, F_2) :- append(F_1, NN, F_2).$$

- **Profundidade**: a fronteira é uma pilha, ou seja, os novos nós entram no início da lista.

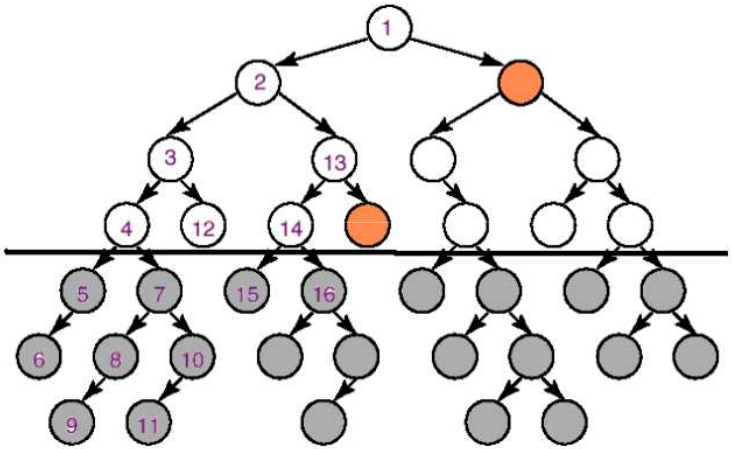
$$add\_to\_frontier(NN, F_1, F_2) :- append(NN, F_1, F_2).$$

ATENÇÃO: Evitar Estados Repetidos

- Não retornar ao estado de onde estamos vindo. Não gere nenhum sucessor de um nó que seja seu pai.
- Não criar caminhos com ciclos. Não gere nenhum sucessor de um nó que seja seu ancestral.
- Não gere qualquer estado que já foi gerado antes. Todos os estados precisam estar armazenados.

Busca em Profundidade Limitada

Busca em Profundidade com limite de profundidade l.
Nós com profundidade l não possuem sucessores.



$search([Node \mid \_]) :- is\_goal(Node).$

$search([Node \mid F_1]) :-$
$\quad depth(Node) < l,$
$\quad neighbors(Node, NN),$
$\quad add\_to\_frontier(NN, F_1, F_2),$
$\quad search(F_2).$

$search([Node \mid F_1]) :- search(F_1).$

Busca em Profundidade Iterativa

Estabelece o limite de profundidade iterativamente. Método indicado quando o espaço de busca é muito grande e a profundidade da solução não é conhecida.

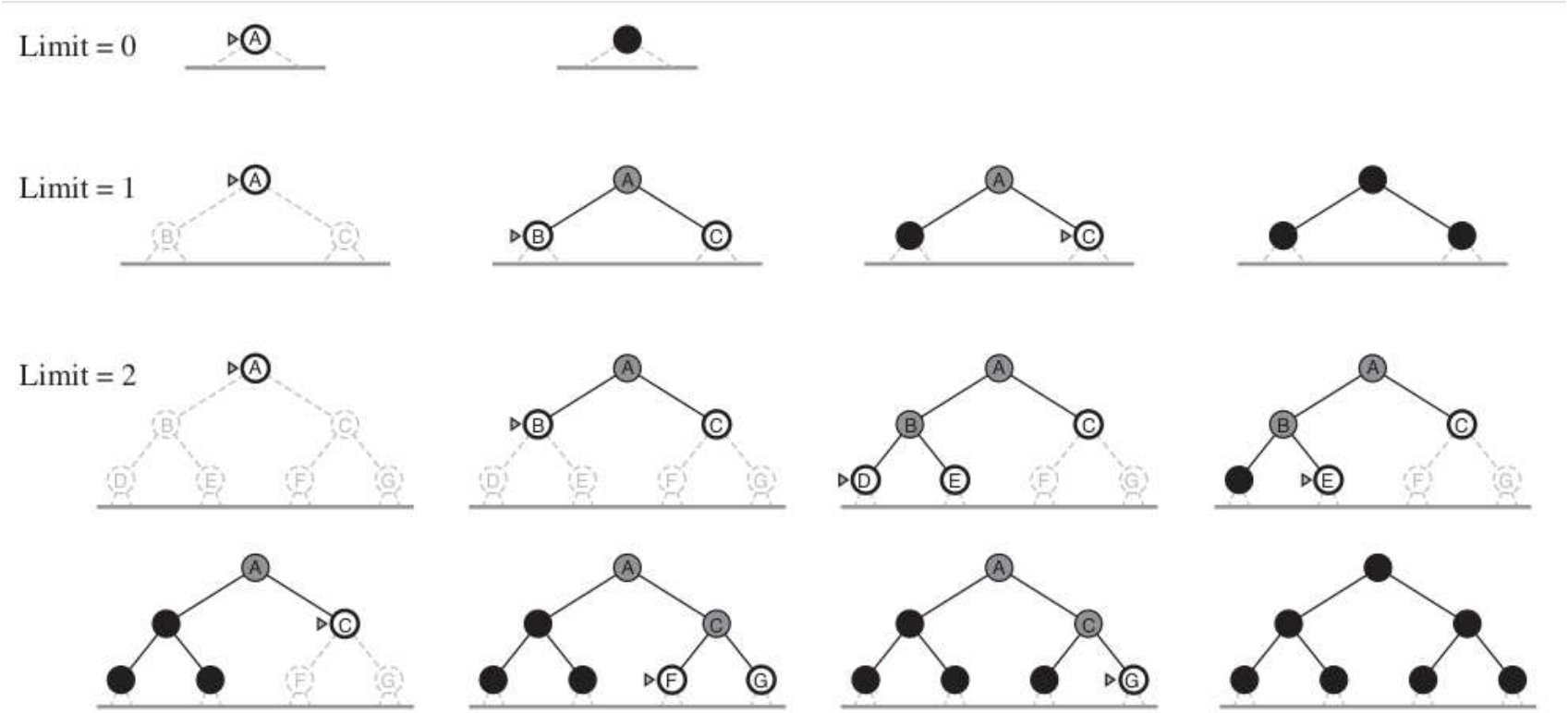


Figura: Fonte: Artificial Intelligence: A Modern Approach - Russell and Norvig

Busca de Custo Uniforme

Expande o nó ainda com menor custo. $g(n)$ (custo do nó inicial até o nó n).

$$add\_to\_frontier(NN, F_1, F_3) \text{ :- } append(F_1, NN, F_2), sort\_by\_g(F_2, F_3).$$

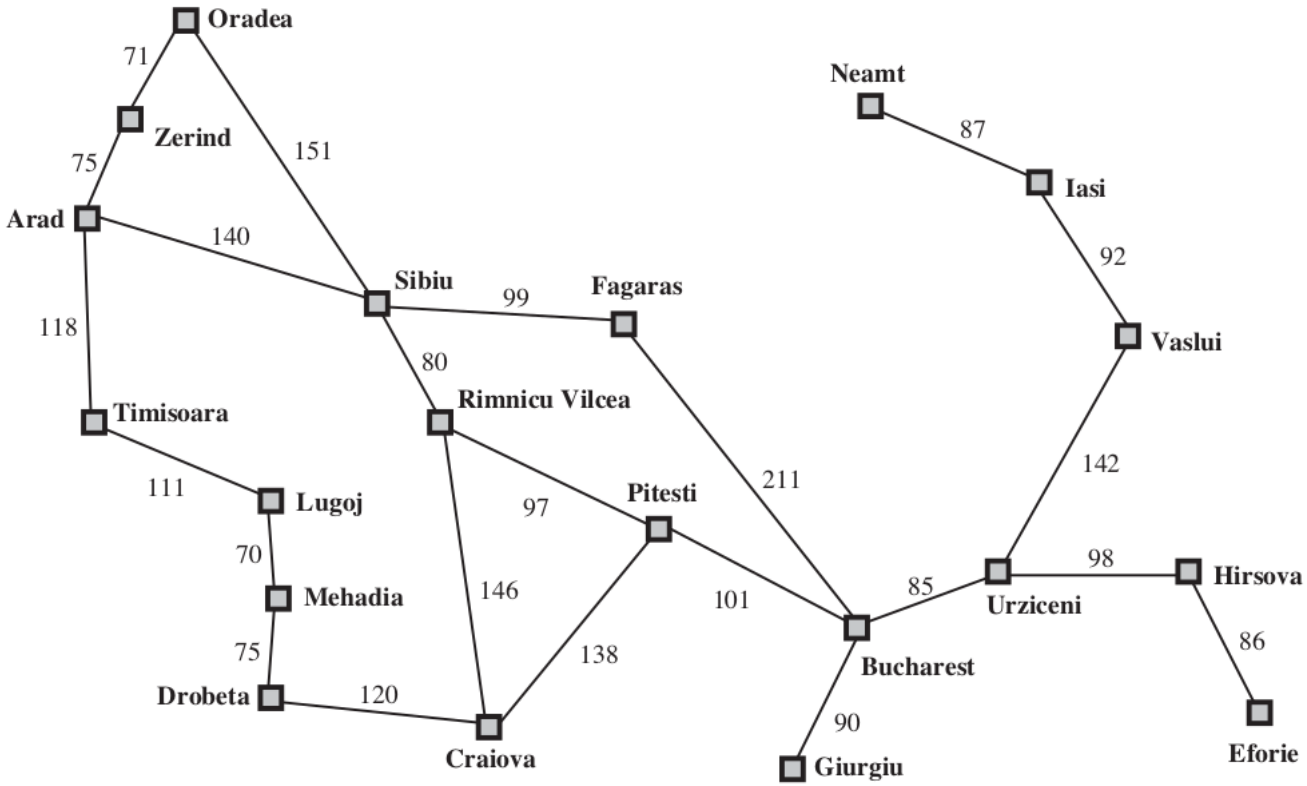
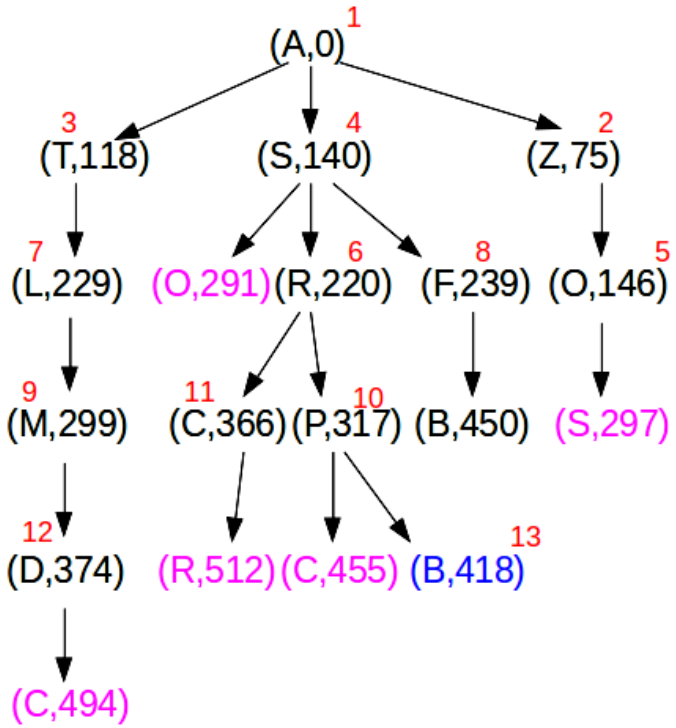


Figura: Fonte: Artificial Intelligence: A Modern Approach - Russell and Norvig

Busca de Custo Uniforme

Expande o nó ainda não expandido com menor custo.



[(A,0)]

[(Z,75), (T,118), (S,140)]

[(T,118), (S,140), (O,146)]

[(S,140), (O,146), (L,229)]

[(O,146), (R,220), (L,229), (F,239)]

[(R,220), (L,229), (F,239)]

[(L,229), (F,239), (P,317), (C,366)]

[(F,239), (M,299), (P,317), (C,366)]

[(M,299), (P,317), (C,366), (B,450)]

[(P,317), (C,366), (D,374), (B,450)]

[(C,366), (D,374), (B,418), (B,450)]

[(D,374), (B,418), (B,450)]

[(B,418), (B,450)]

Busca Bidirecional

Busca nos dois sentidos (Inicial-Final e Final-Inicial).

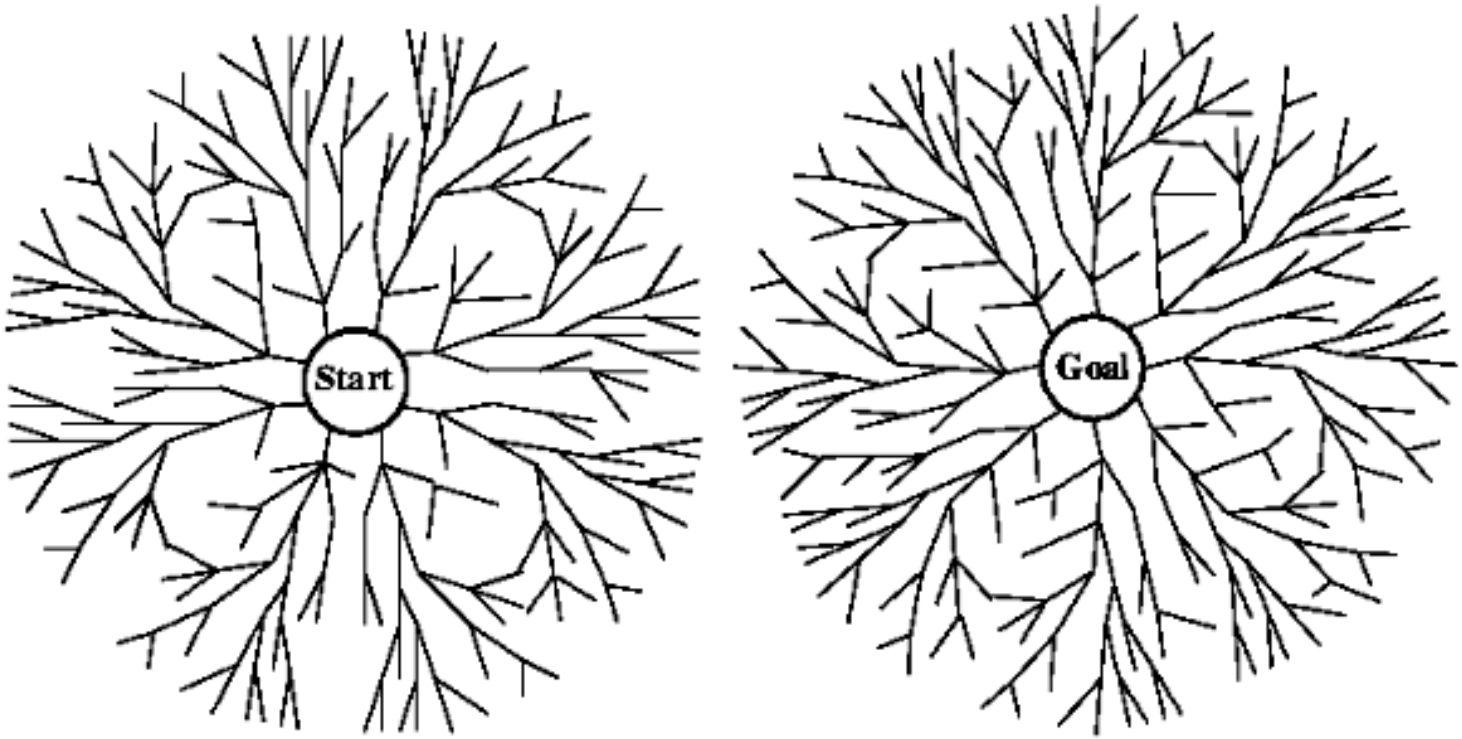


Figura: Fonte: Artificial Intelligence: A Modern Approach - Russell and Norvig

Inteligência Artificial
Aula 5 - vídeo 3 - Buscas Não Informadas

João C. P. da Silva

Dept. Ciência da Computação - UFRJ

11 de setembro de 2020