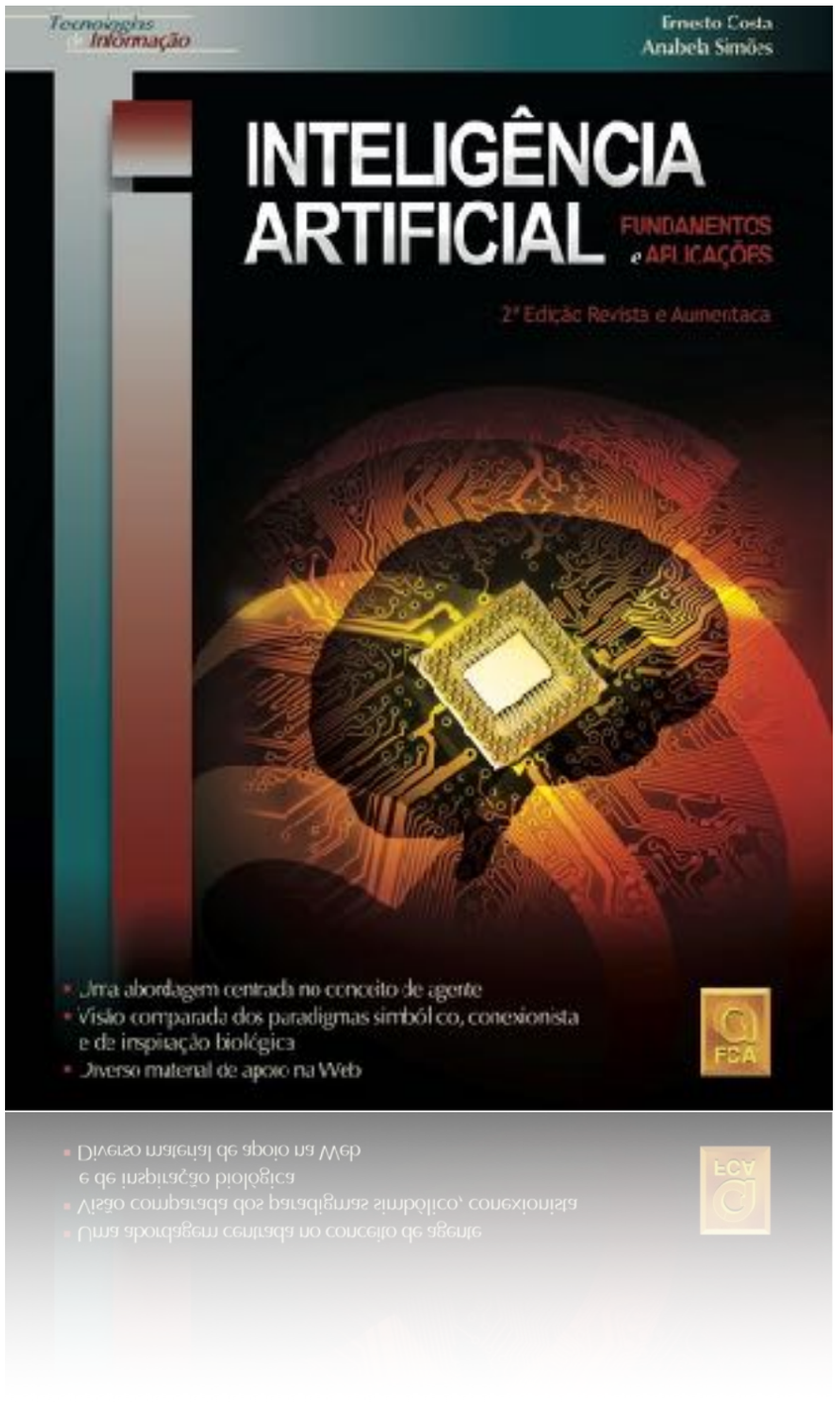
Agentes de Procura

IIA / FIA
2024/2025

Bibliografia

IAFA

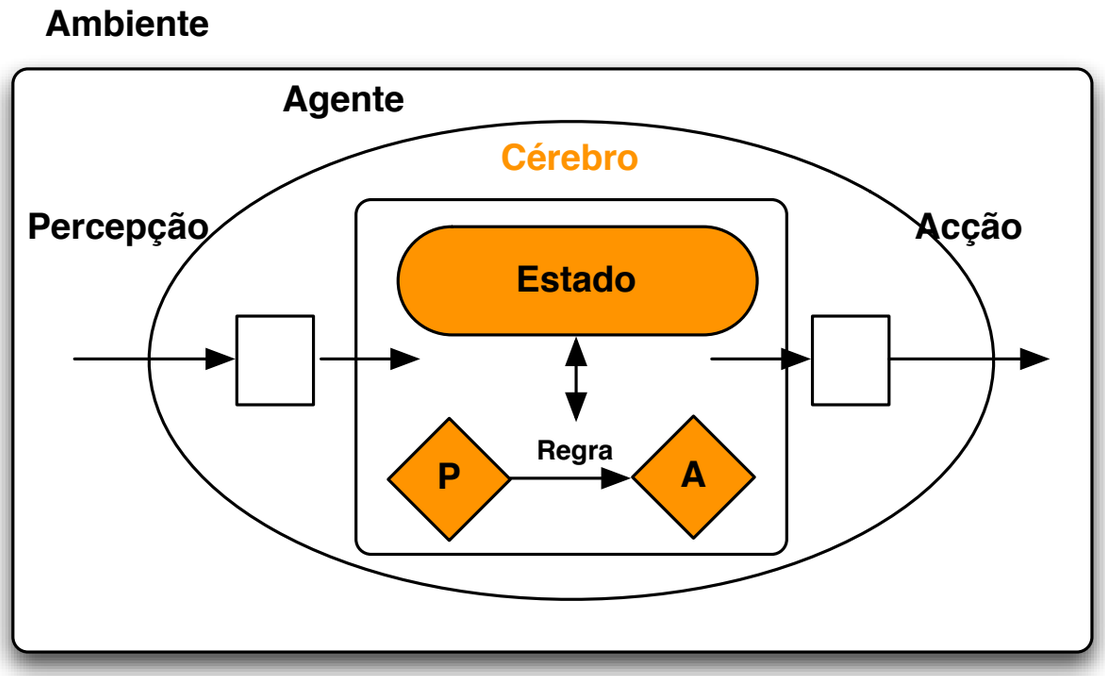
Cap. 3


Tecnologias de Informação
Ernesto Costa
Anabela Simões
INTELIGÊNCIA ARTIFICIAL
FUNDAMENTOS e APLICAÇÕES
2ª Edição Revista e Aumentada
• Uma abordagem centrada no conceito de agente
• Visão comparada dos paradigmas simbólico, conexionista e de inspiração biológica
• Diverso material de apoio na Web
FCA

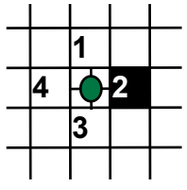
Agentes Reactivos

Limitações

Visão local do meio-ambiente



Estado



Memória

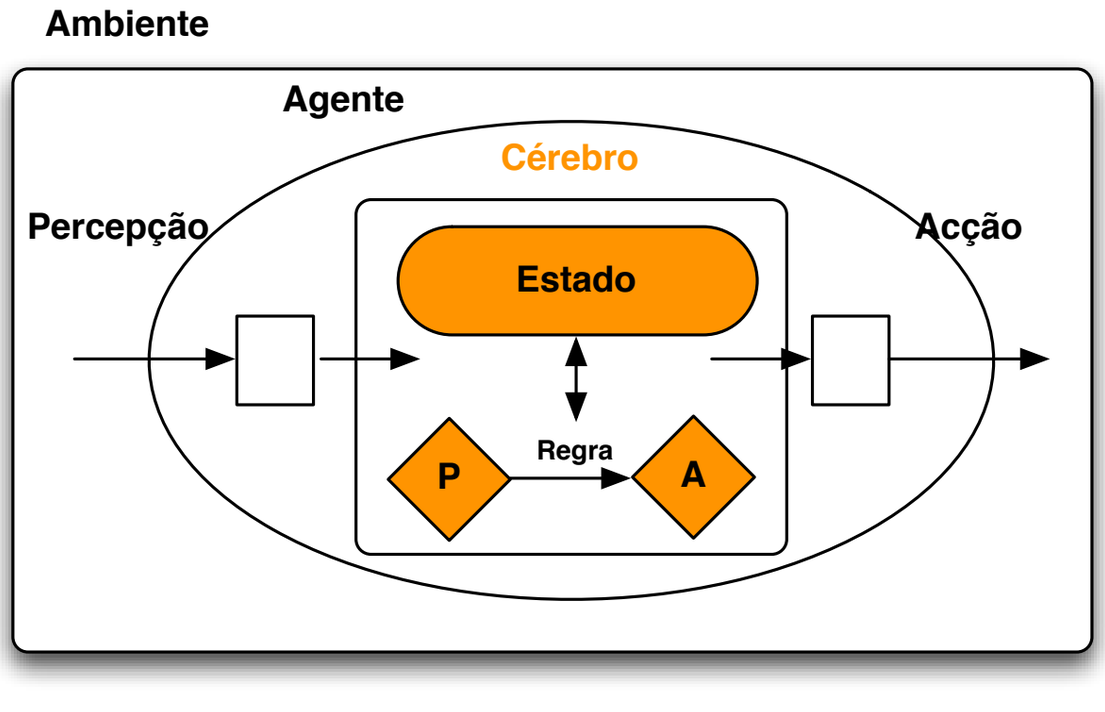
Regras

1.Off,F → A,Off
2.Off, ¬F → E,On
3.On,F → A,Off
4.On, ¬F → D,On

Limitações

Visão local do meio-ambiente

O controlador é fixo



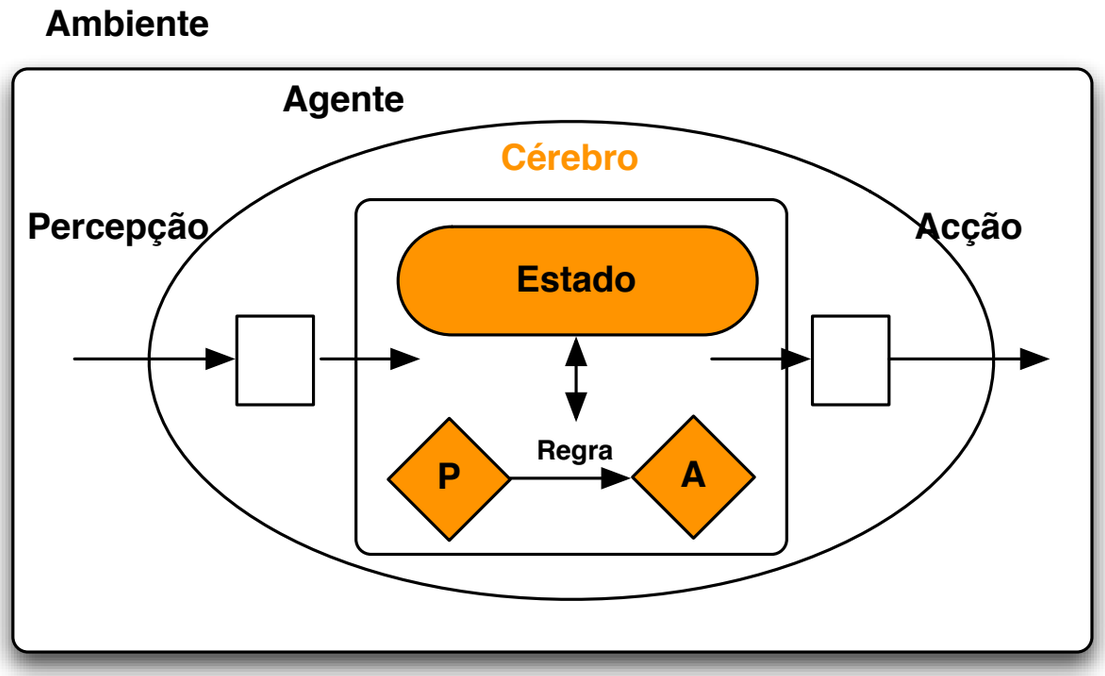
Enquanto puder:
- **analisa** estado
- **determina** regras/comportamento aplicáveis
- **escolhe** regra/comportamento por ordem
- **aplica** regra/comportamento

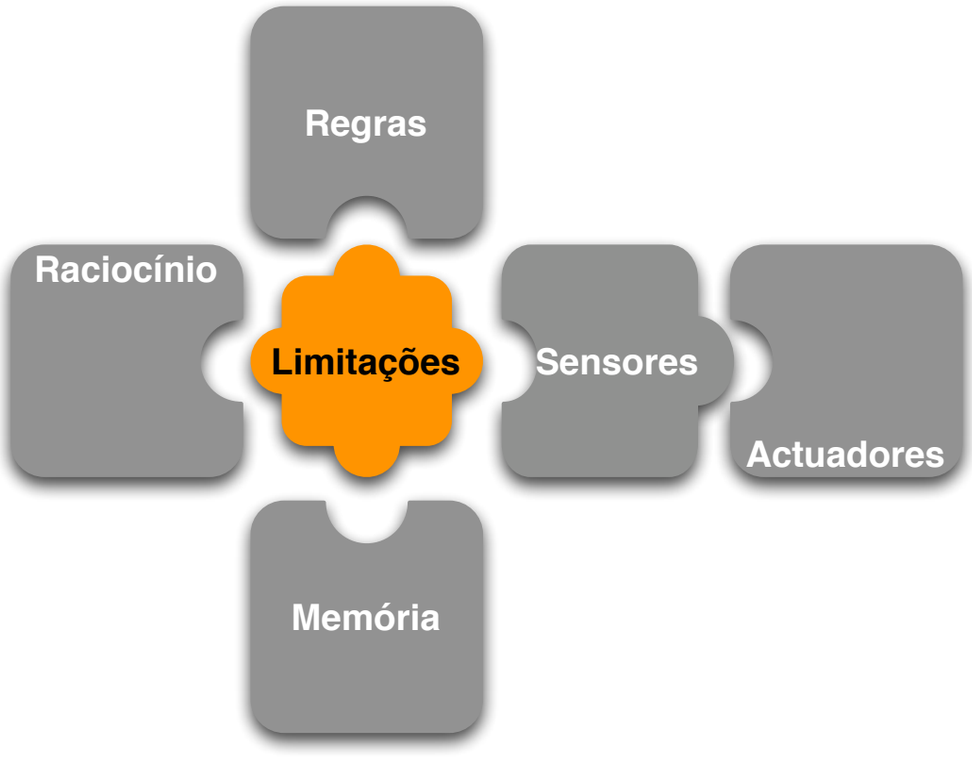
Limitações

Visão local do meio-ambiente

O controlador é fixo

Que tipo de tarefas podem resolver?

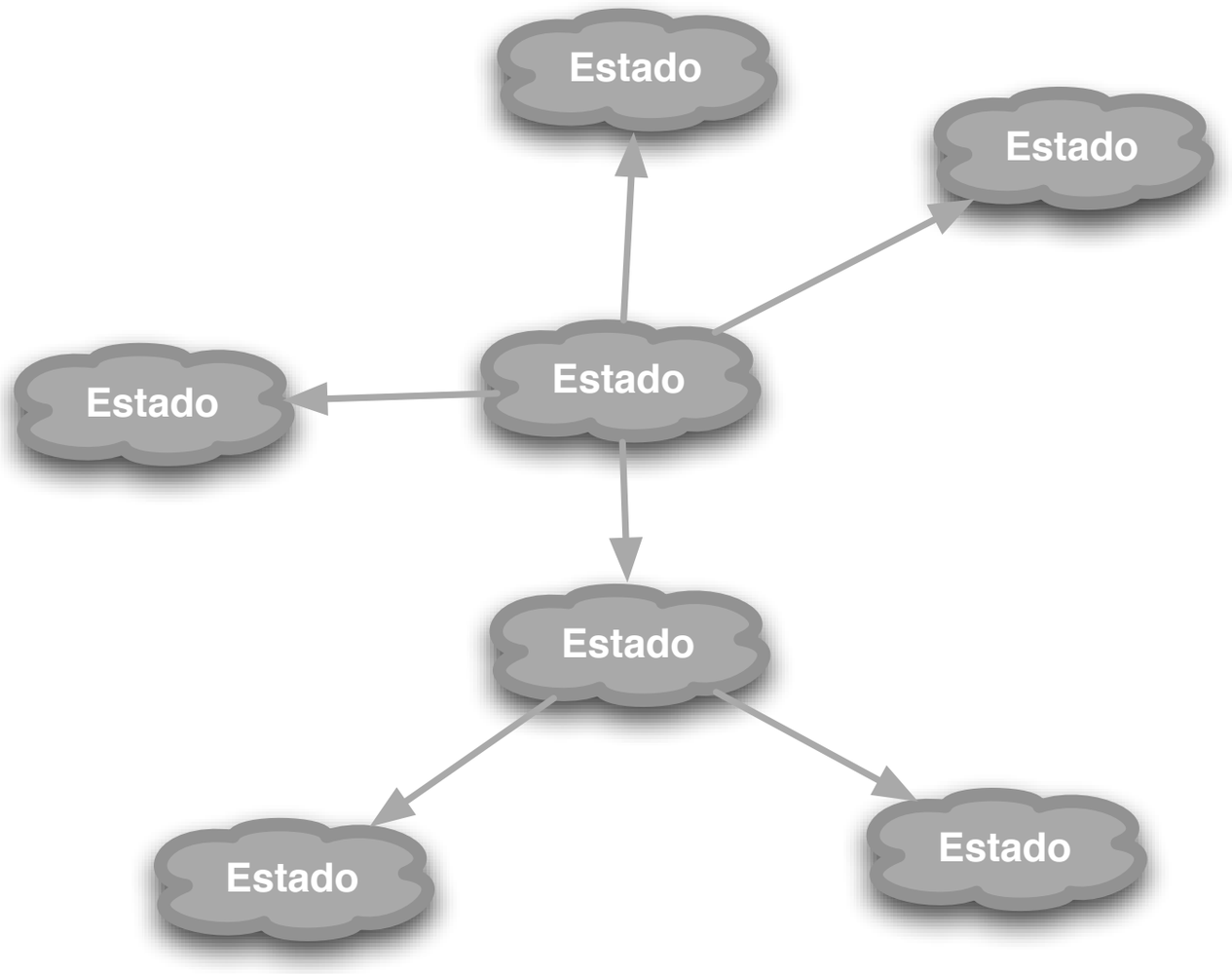




Agentes de Procura

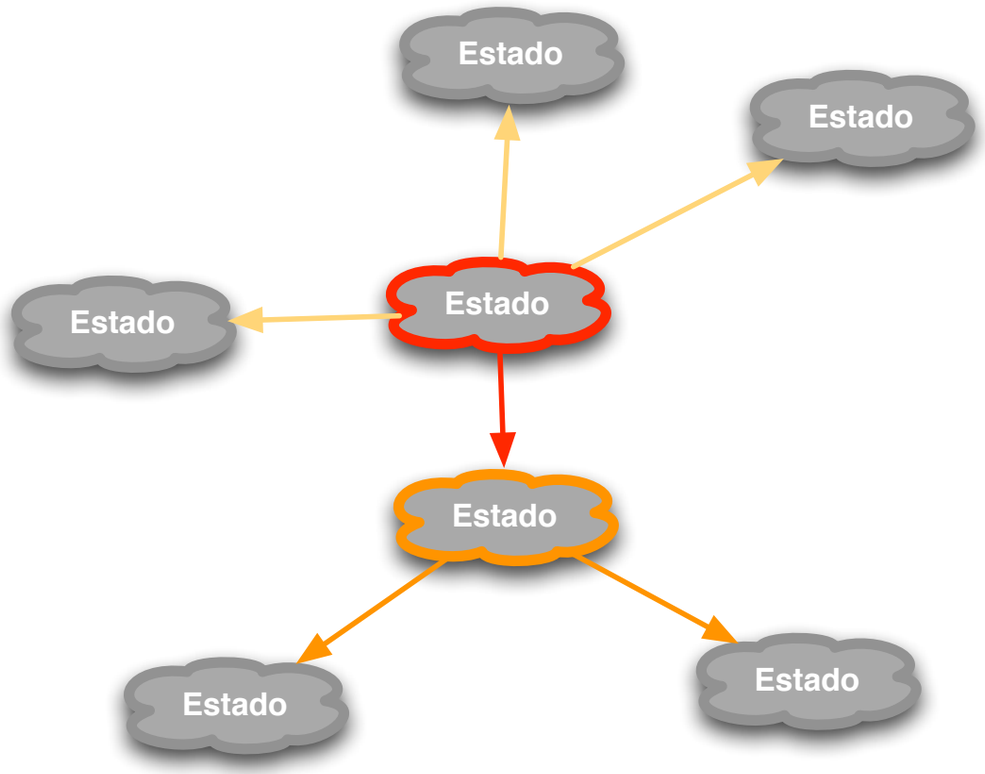
Agentes de Procura

Mais memória
Estados
Relações entre estados



Agentes de Procura

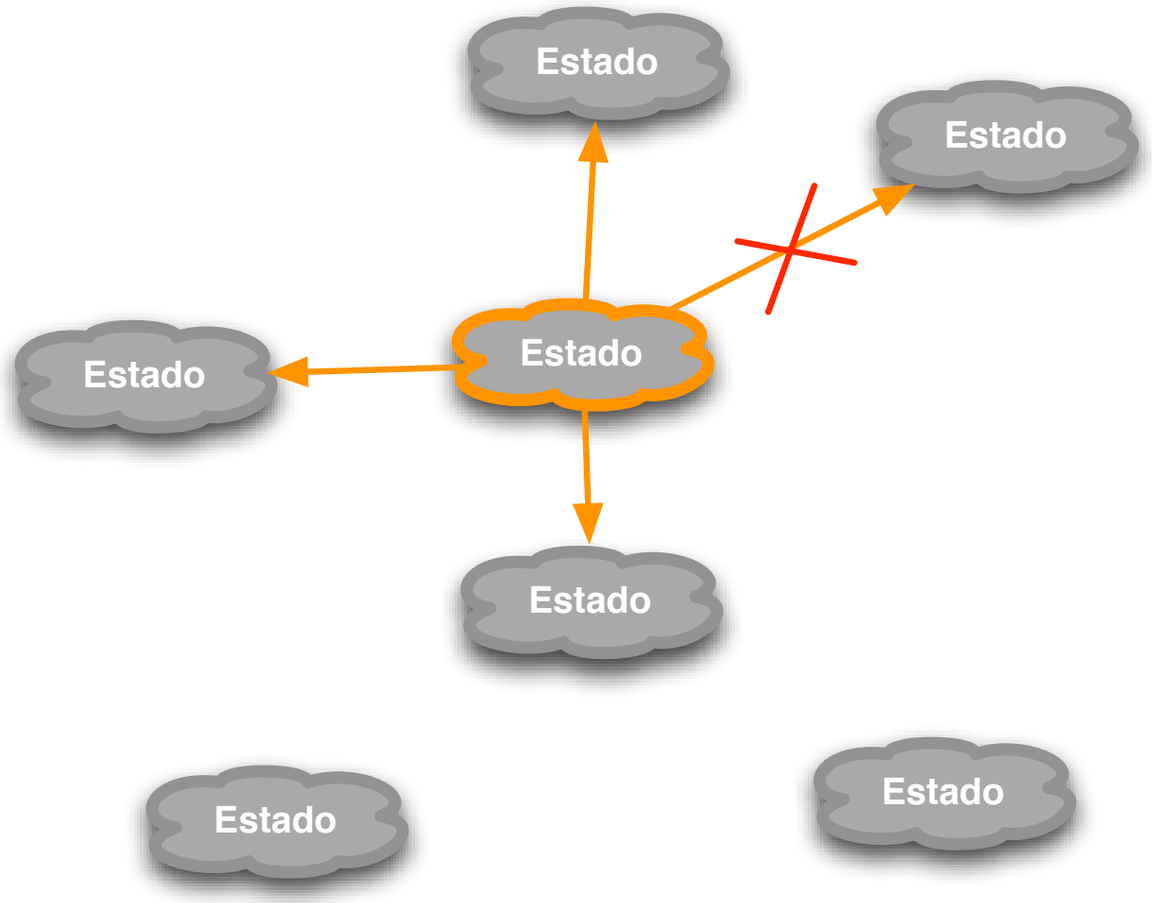
Controlador mais sofisticado
Simulação / Antevisão



Enquanto não resolvido e tiver alternativas:
analisa estado
determina regras aplicáveis
aplica de modo simulado todas as regras
escolhe controladamente um estado resultante

Agentes de Procura

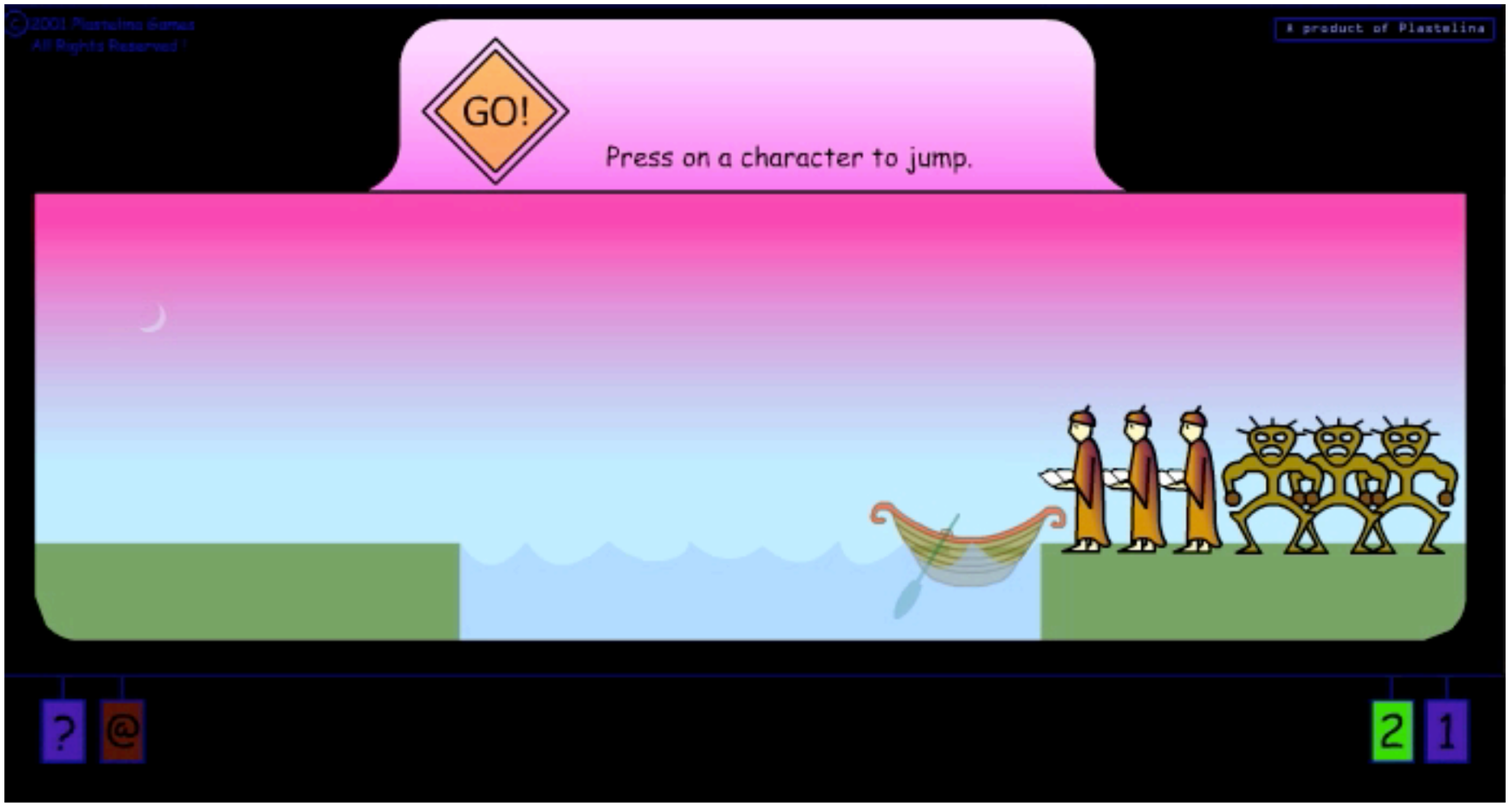
Restrições
Operadores de Mudança de estado
Limitações



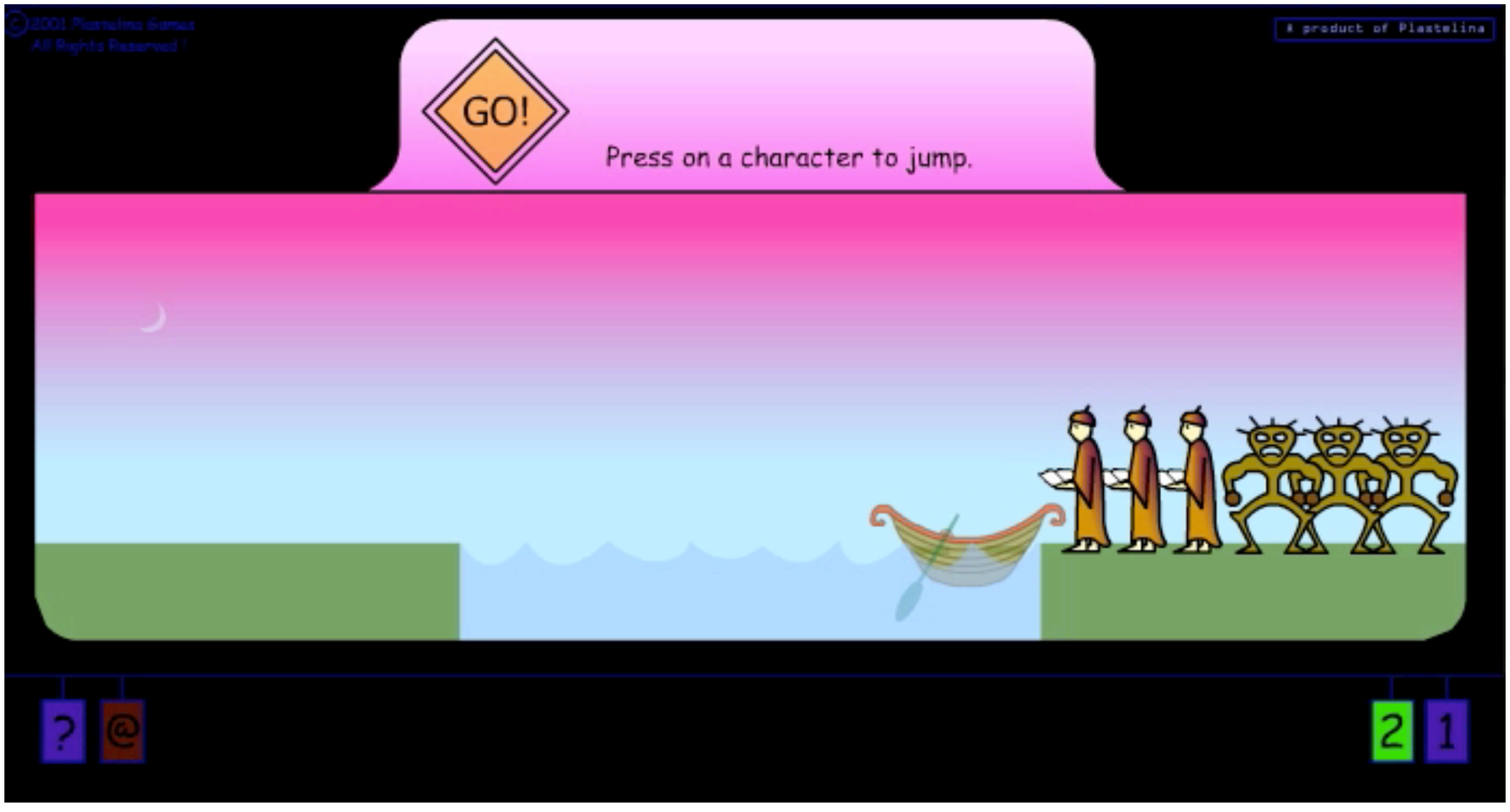
Agentes de Procura

Resolução de Problemas

Missionários e Canibais

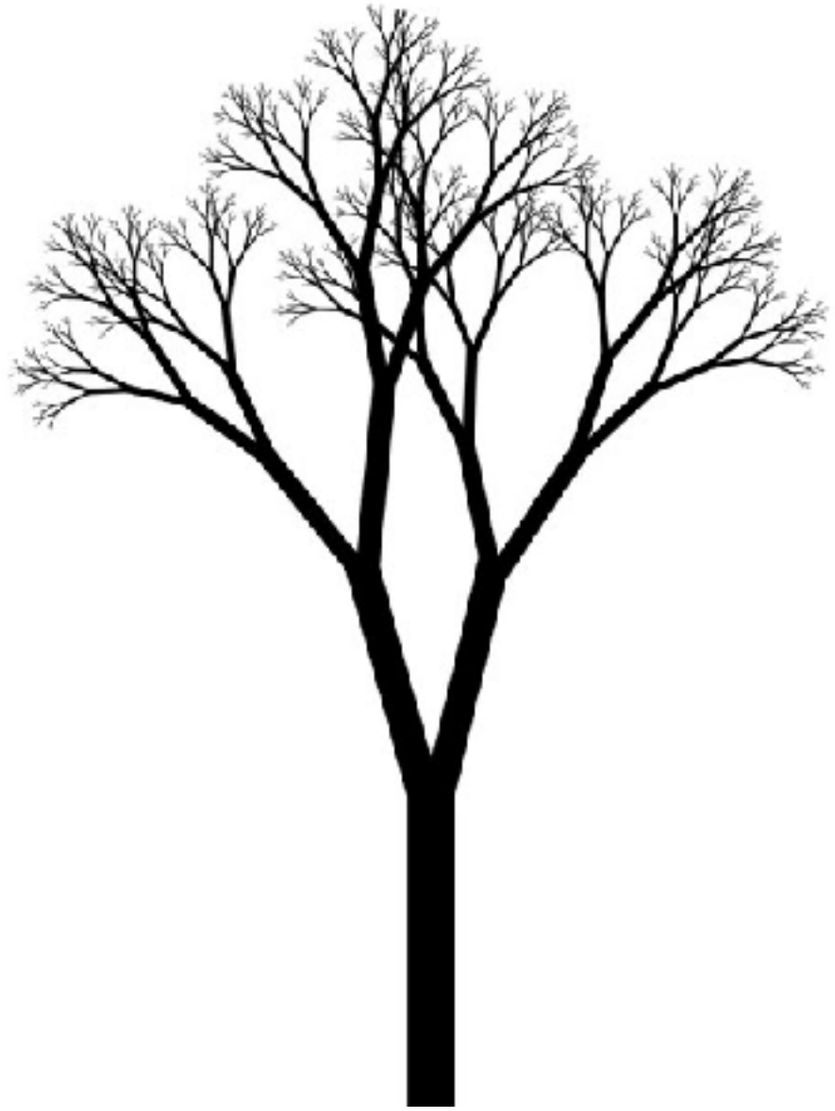


Missionários e Canibais



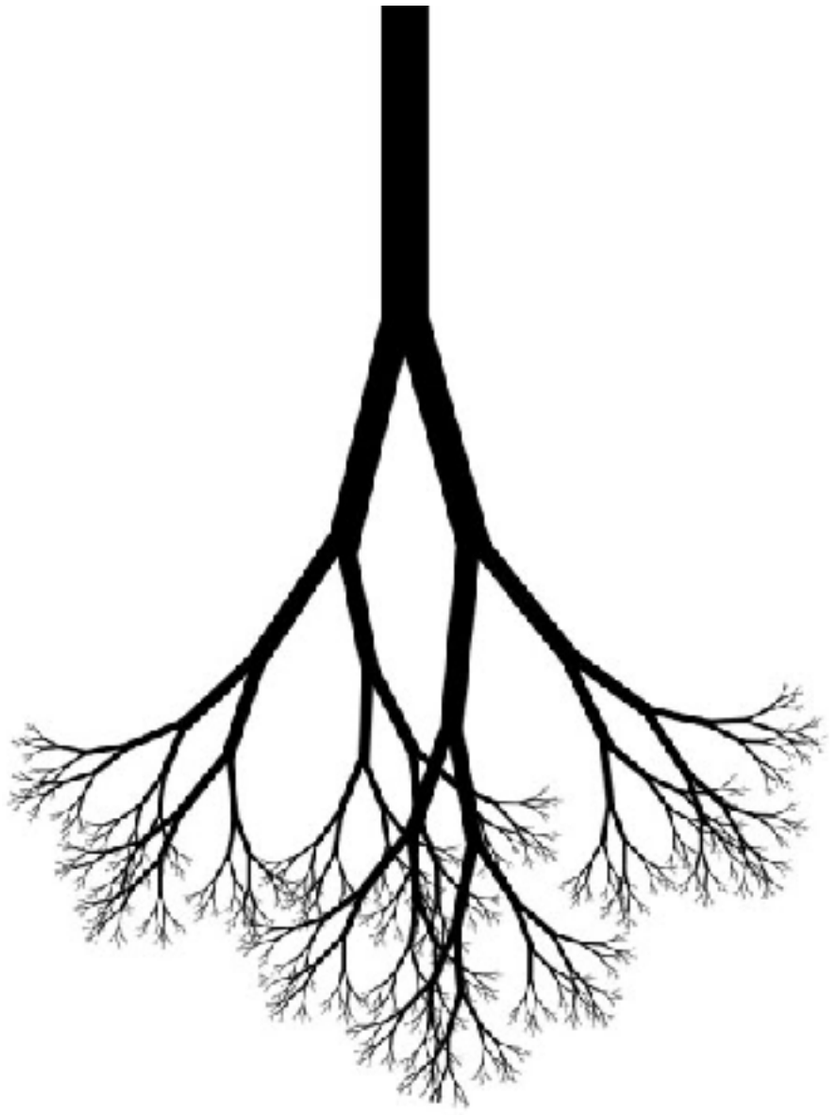
Conceitos Comuns

Estado
Operadores de Mudança de Estado
Restrições
Árvore/Espaço de procura



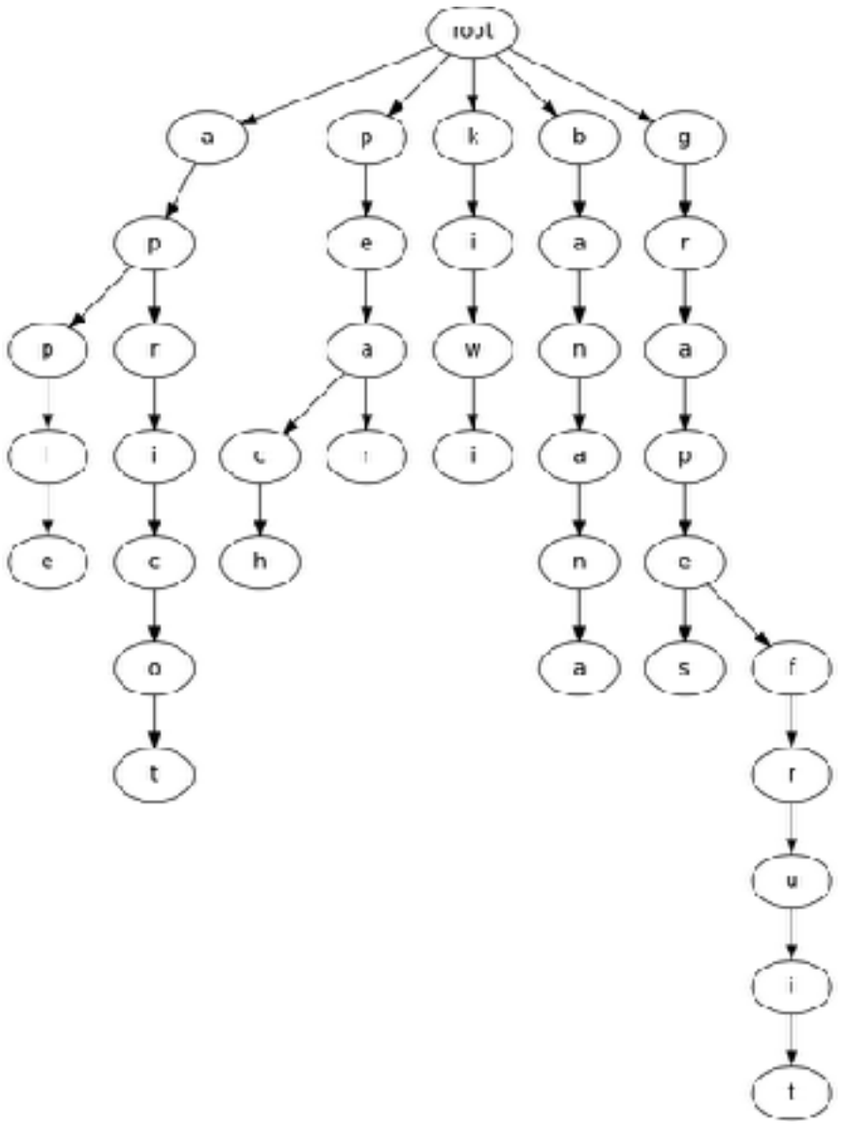
Conceitos Comuns

Estado
Operadores de Mudança de Estado
Restrições
Árvore/Espaço de procura



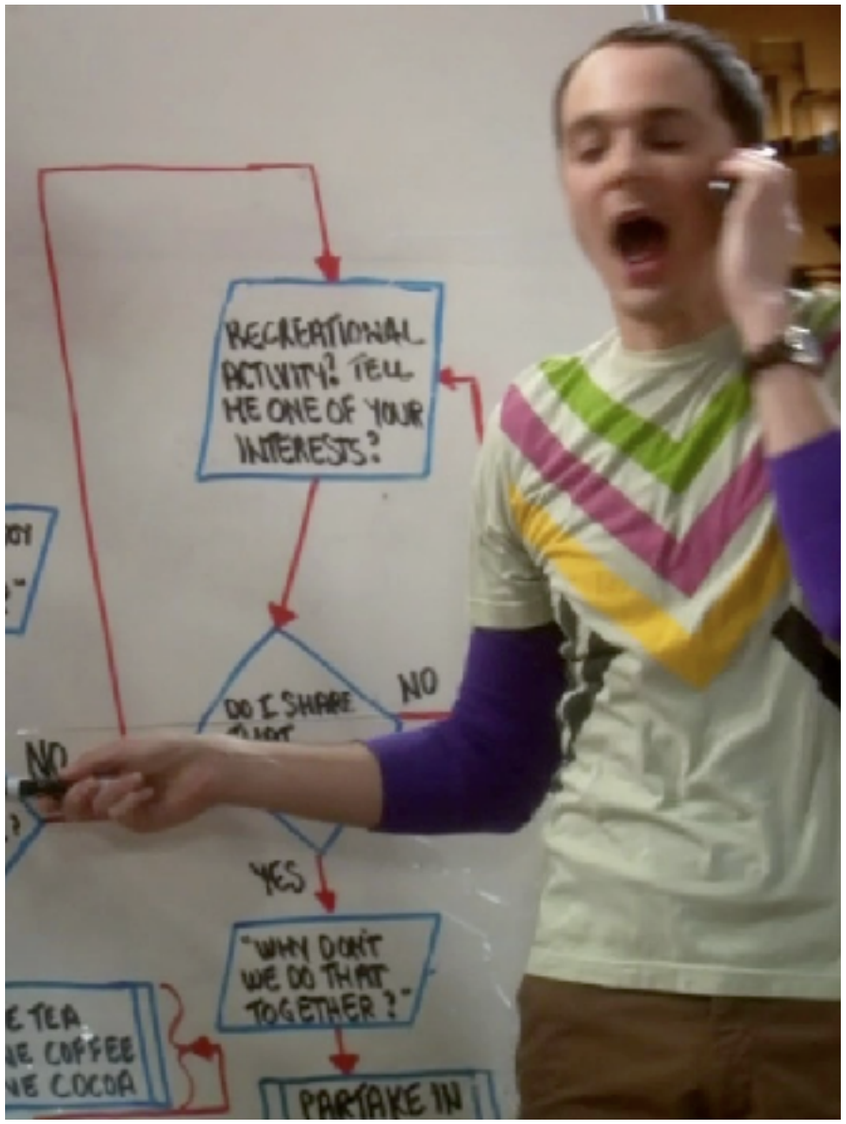
Conceitos Comuns

Estado
Operadores de Mudança de Estado
Restrições
Árvore/Espaço de procura



O que difere?

Algoritmo de procura
Resultado

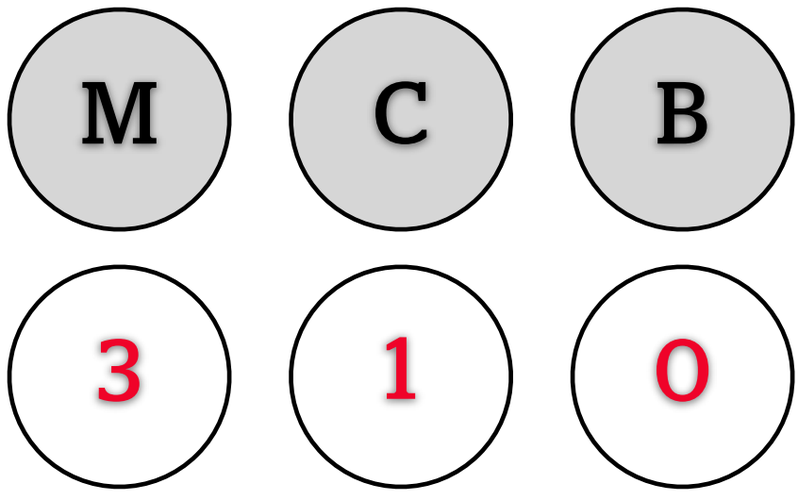


Estado

Deve identificar de forma completa cada situação

Missionários e Canibais

Missionários na direita
Canibais na margem direita
Posição do barco



Operadores de Mudança de Estado

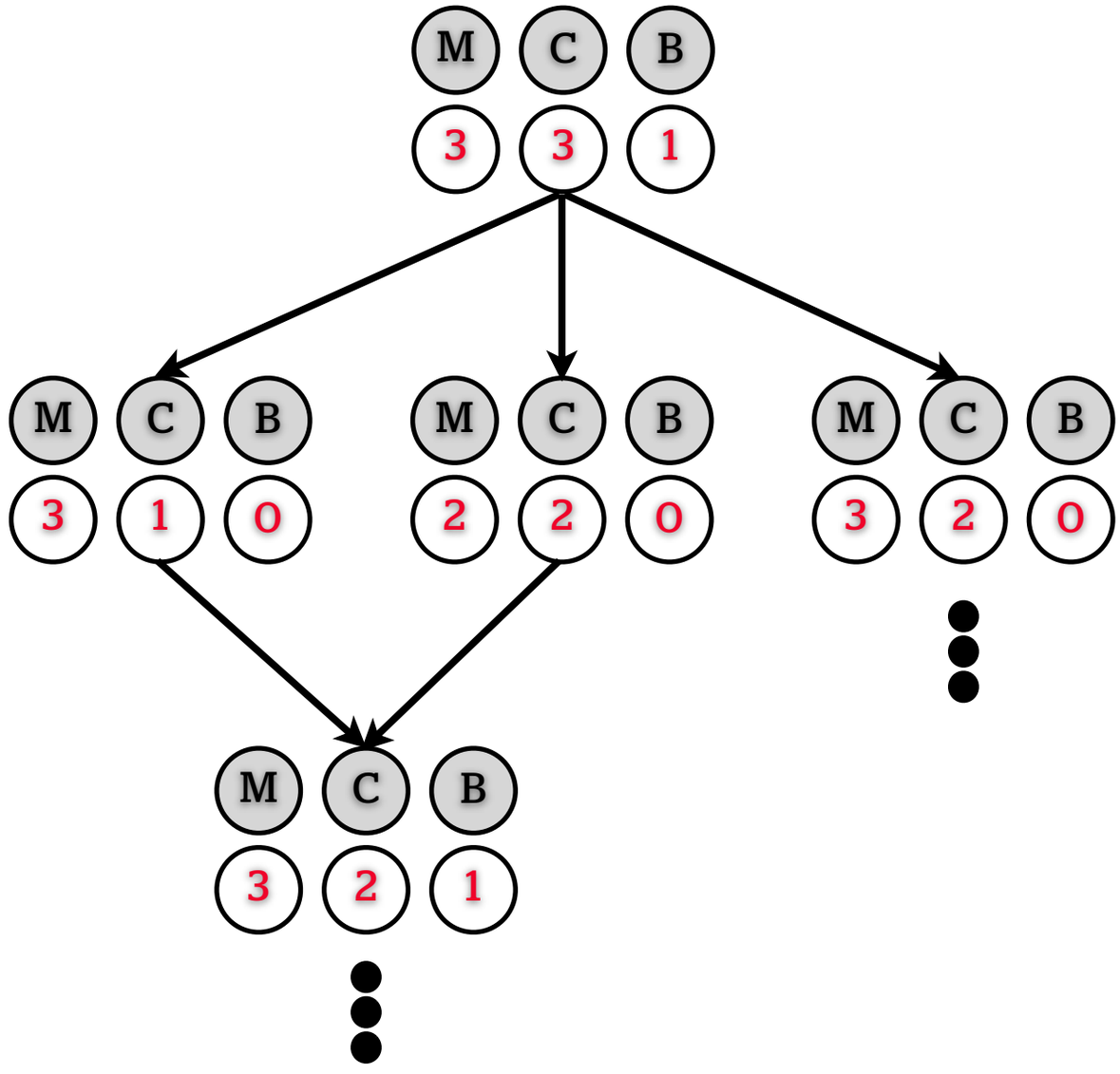
Determinam todas as acções possíveis a partir de cada estado

Missionários e Canibais

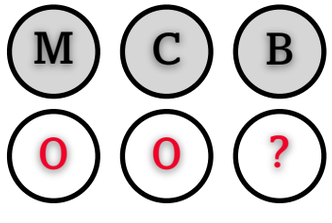
Transportar para a margem oposta:
um missionário
um canibal
dois missionários
dois canibais
um missionário e um canibal
desde que tal não viole as restrições do problema!

Árvore de procura

Resultante da aplicação recursiva dos operadores ao estado inicial



Estado final

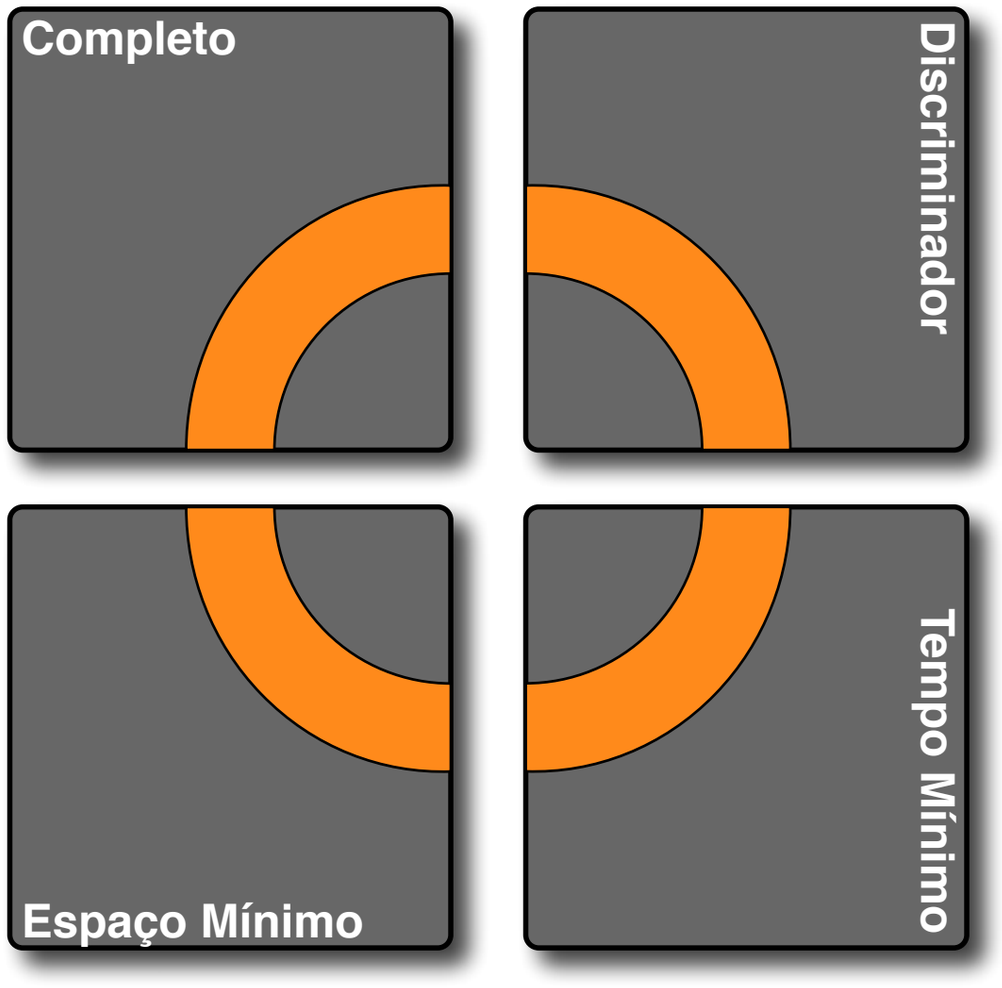


Algoritmo Geral de Procura

Algoritmo Geral de Procura

função ProcuraGeral (problema, estratégia): solução ou falha
1.inicializa a árvore de procura com o estado inicial do problema
2. repete
  2.1 se não há candidatos para expandir
    então 2.1.1 devolve falha
    fim_de_se
  2.2 escolhe um estado na fronteira para expansão, de acordo com a estratégia
  2.3 se o estado contém o objectivo
    então 2.3.1 devolve a solução correspondente
    senão 2.3.2 expande o estado e acrescenta à árvore de procura os seus sucessores, de acordo com a estratégia
    fim_de_se
fim_de_repete
fim_de_função

Desempenho dos Agentes de Procura



Estratégias e Soluções

Estratégia

Completa?

Discriminadora?

Complexidade Temporal?

Complexidade Espacial?

Tipos de Solução

A sequência de acções

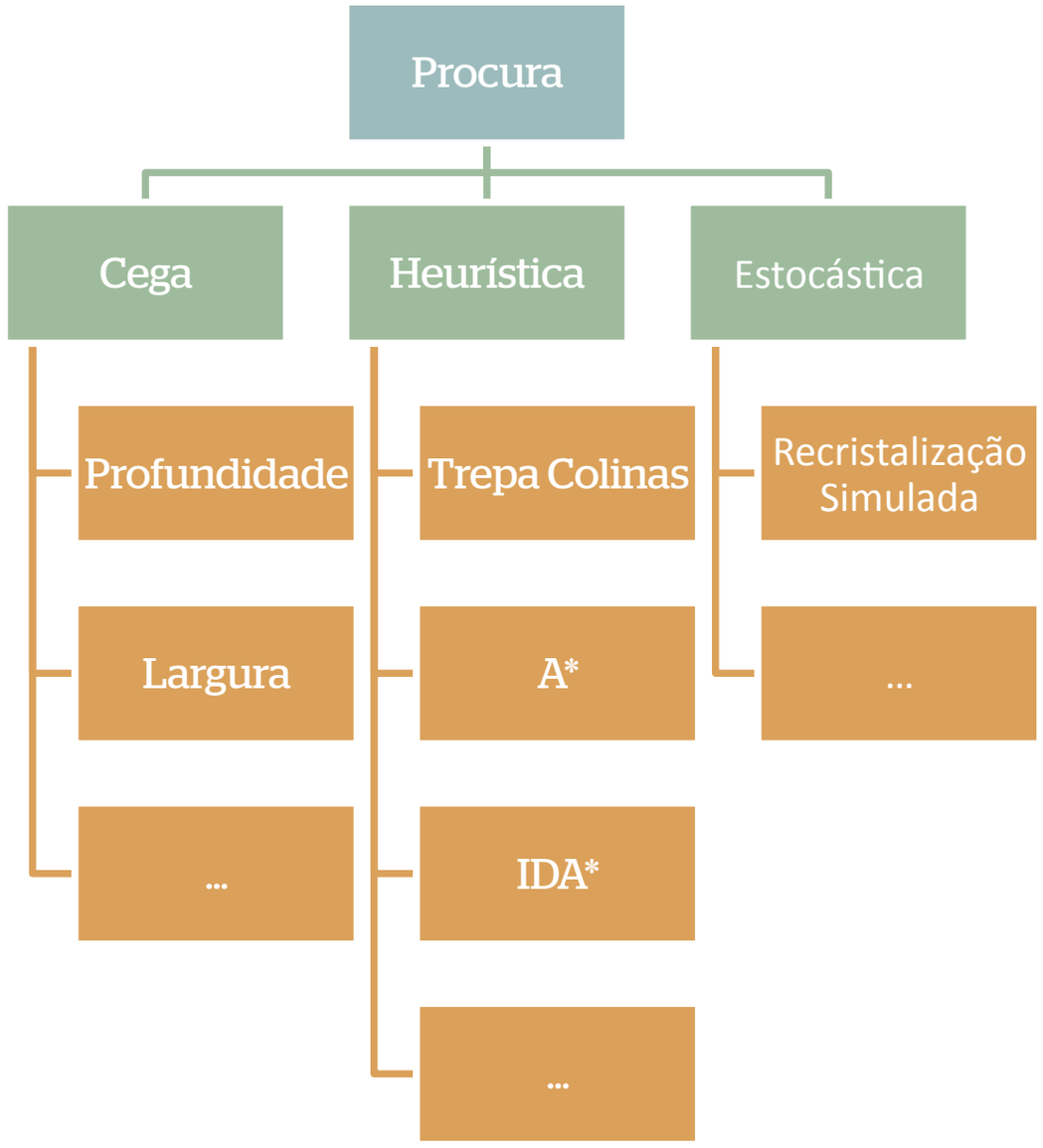
A solução

Uma solução

A melhor solução

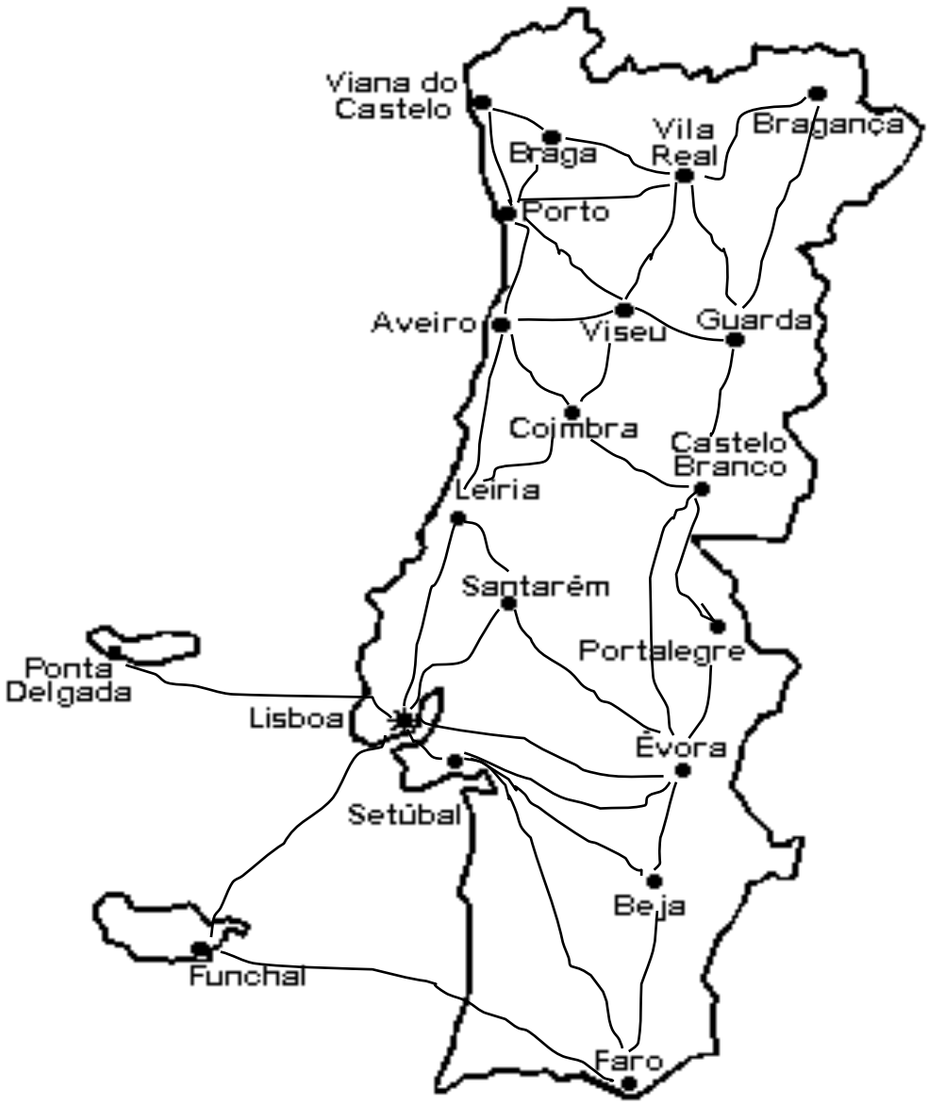
Tipos de Agentes de Procura

Estratégias



Um Problema Clássico

Encontrar um caminho



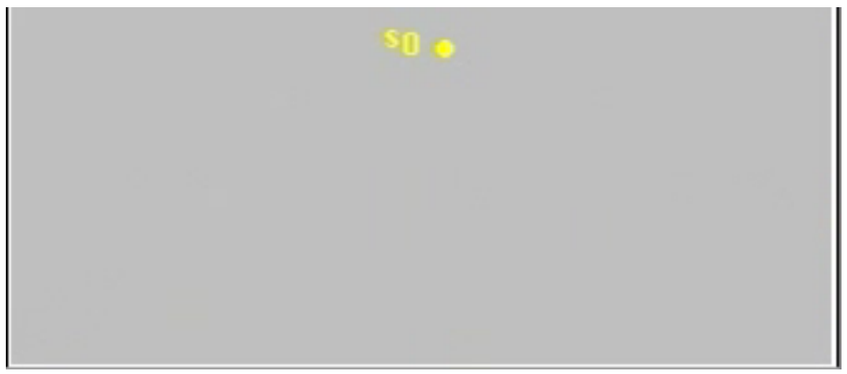
Aveiro	Porto (68)	Viseu (95)	Coimbra(68)	Leiria (115)
Braga	Viana C.(48)	V. Real (106)	Porto (53)	
Bragança	V. Real (137)	Guarda (202)		
Beja	Évora (78)	Faro (152)	Setúbal (142)	
C. Branco	Coimbra (159)	Guarda (106)	Portalegre (80)	Évora(203)
Coimbra	Viseu (96)	Leiria (67)		
Évora	Lisboa (150)	Santarém (117)	Portalegre (131)	Setúbal (103)
Faro	Setúbal (249)	Lisboa (299)		
Guarda	V. Real (157)	Viseu (85)		
Leiria	Lisboa (129)	Santarém (70)		
Lisboa	Santarém (78)	Setúbal (50)		
Porto	V. Castelo (71)	V. Real (116)	Viseu (133)	
V. Real	Viseu (110)			

Coimbra Leiria Lisboa Setúbal Faro

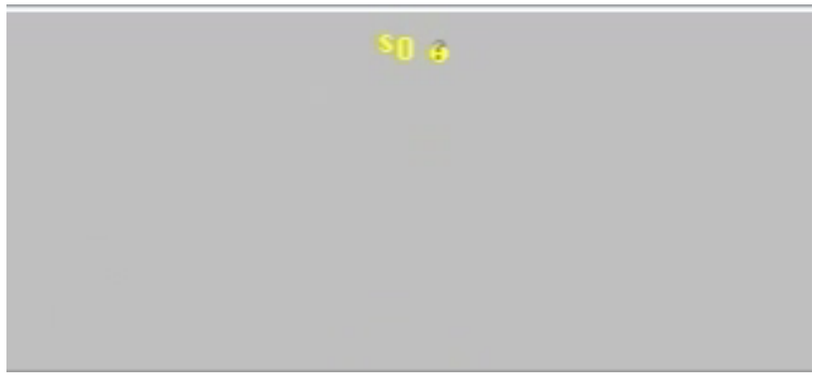
Procura Cega

Procura Cega

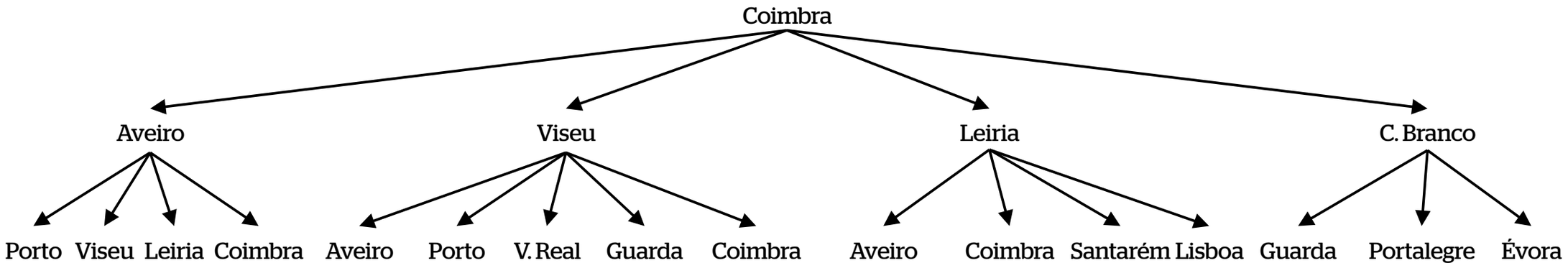
Largura



Profundidade

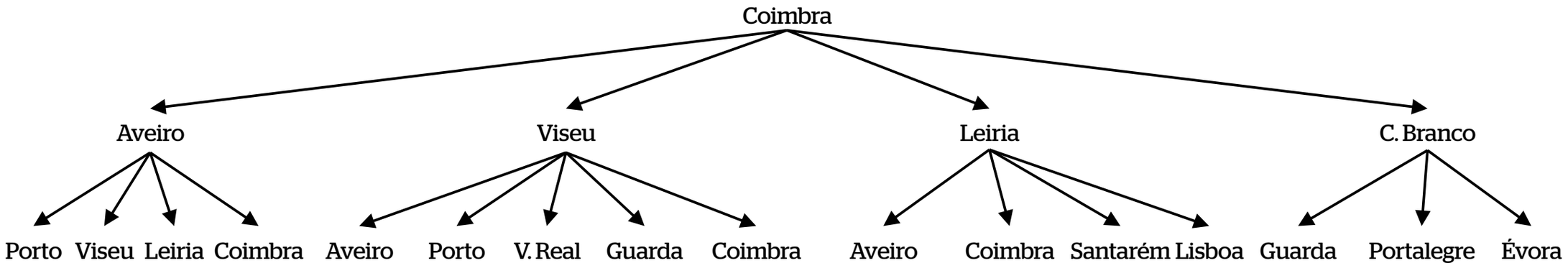


Largura Primeiro



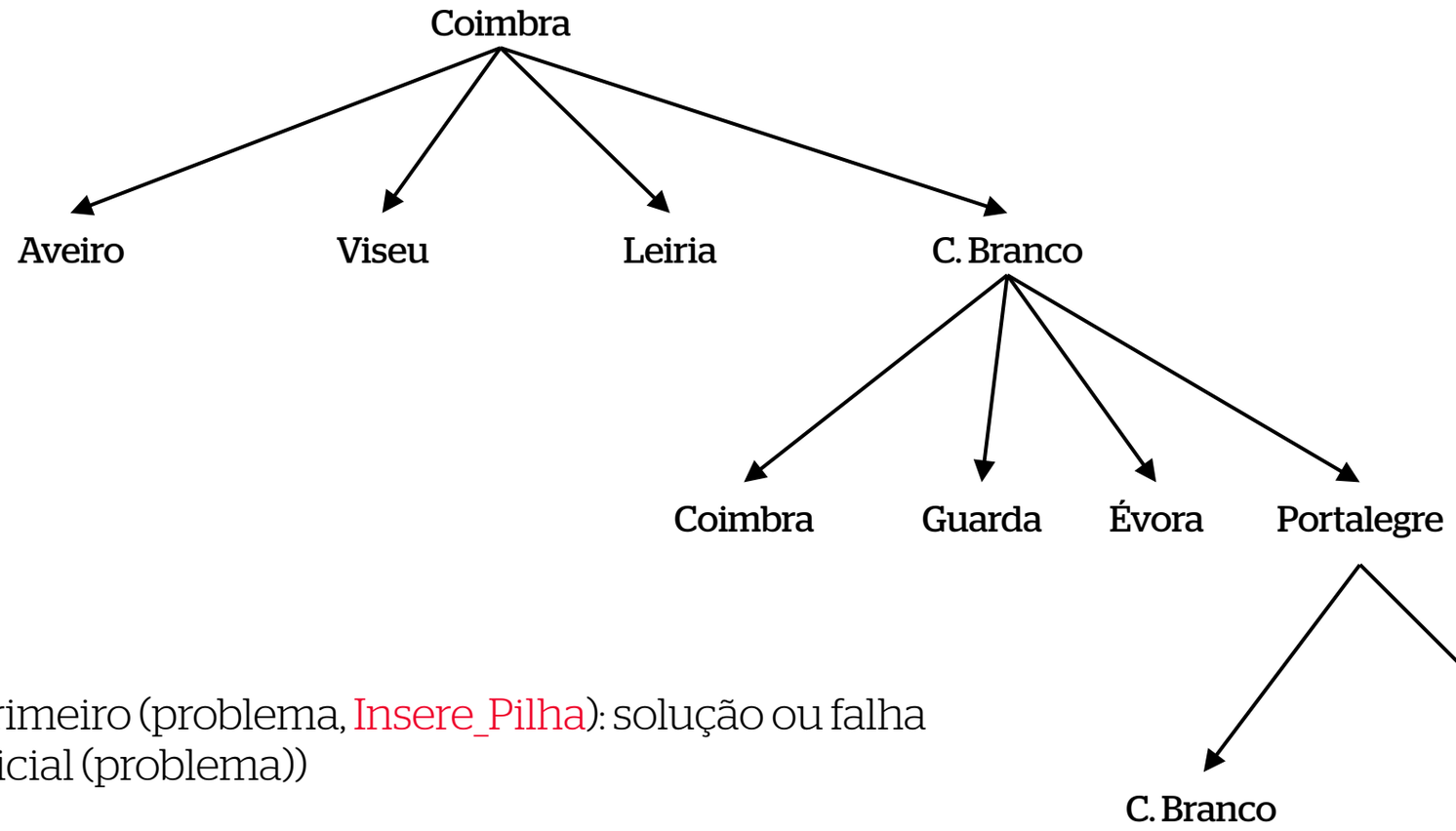

```
função ProcuraLarguraPrimeiro (problema, Insere_Fila): solução ou falha
1. 1 nós ← Faz_Fila (Estado_Inicial (problema))
2. repete
  2.1 se Vazia_Fila(l_nós)
      então 2.1.1 devolve falha
      fim_de_se
  2.2 nó ← Retira_Fila(l_nós)
  2.3 se Teste_Objectivo(nó)
      então 2.3.1 devolve nó
      senão  2.3.2 Insere_Fila(l_nós, Expansão(nó, Operadores(problema)))
      fim_de_se
fim_de_repete
fim_de_função
```

Largura Primeiro



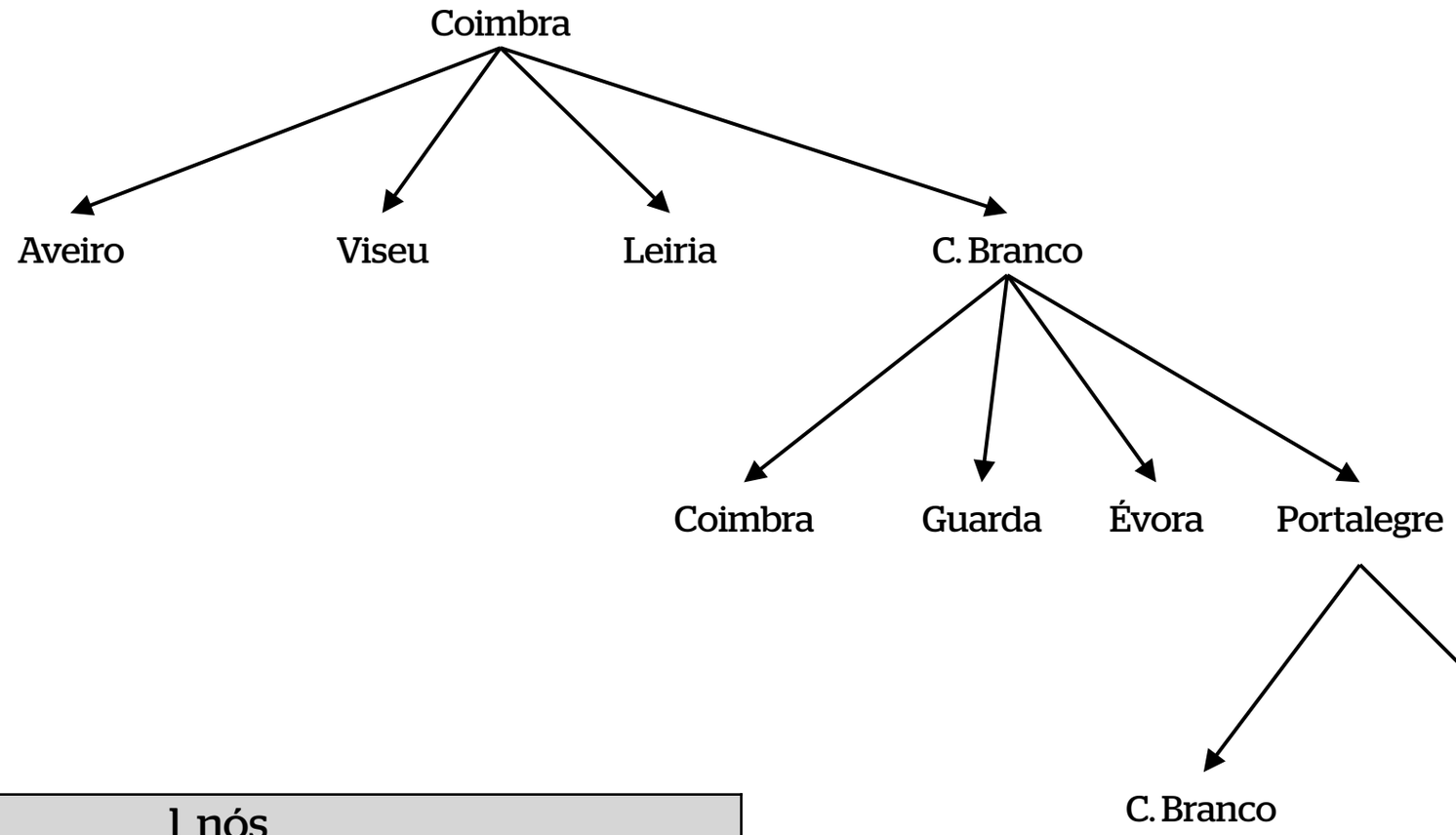
Iteração	l_nós
0	[Coimbra]
1	[Aveiro, Viseu, Leiria, C. Branco]
2	[Viseu, Leiria, C. Branco, Porto, Viseu, Leiria, Coimbra]
3	[Leiria, C. Branco, Porto, Viseu, Leiria, Coimbra, Aveiro, Porto, V. Real, Guarda, Coimbra]
...	...

Profundidade Primeiro



função ProcuraProfundidadePrimeiro (problema, Insere_Pilha): solução ou falha
1.1 nós ← Faz_Pilha (Estado_Inicial (problema))
2. repete
2.1 se Vazia_Pilha(l_nós)
então 2.1.1 devolve falha
fim_de_se
2.2 nó ← Retira_Pilha(l_nós)
2.3 se Teste_Objectivo(nó)
então 2.3.1 devolve nó
senão 2.3.2 Insere_Pilha(l_nós, Expansão(nó, Operadores(problema)))
fim_de_se
fim_de_repete
fim_de_função

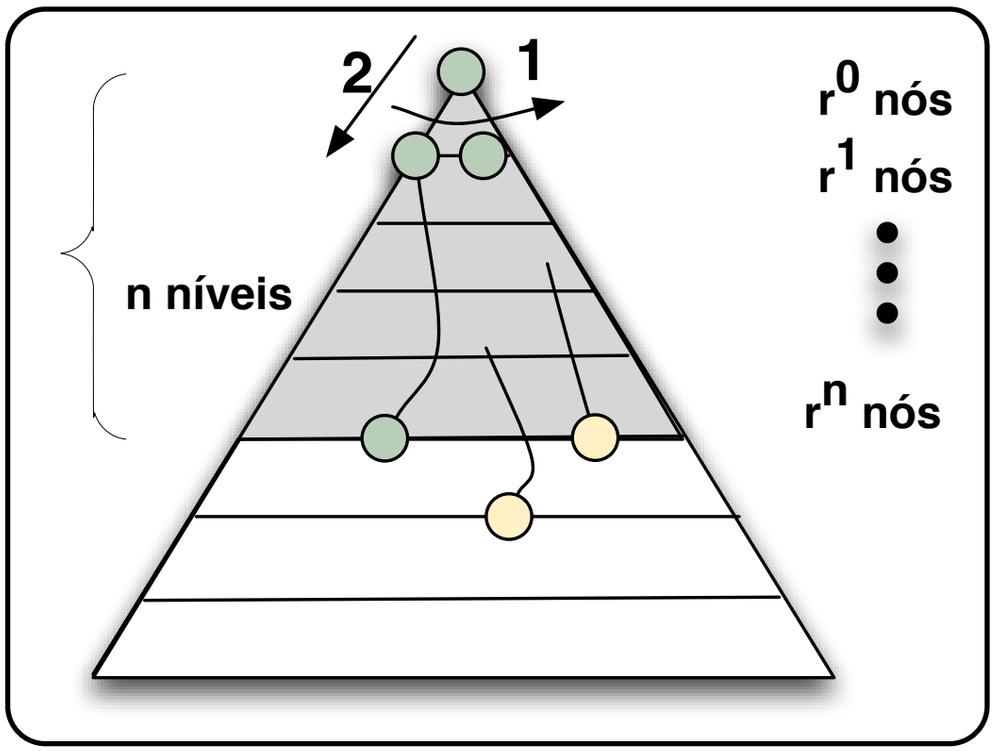
Profundidade Primeiro

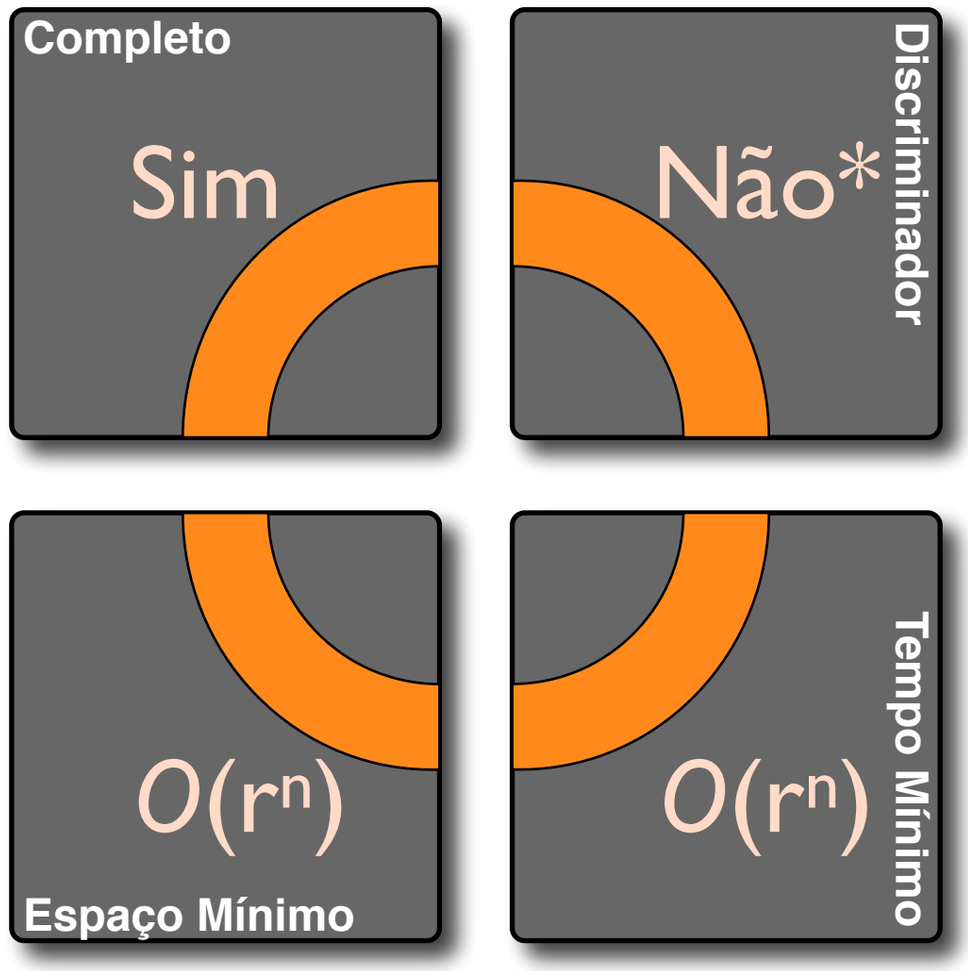


Iteração	l_nós
0	[Coimbra]
1	[C. Branco, Leiria, Viseu, Aveiro]
2	[Portalegre, Évora, Guarda, Coimbra, Leiria, Viseu, Aveiro]
3	[Évora, C. Branco, Évora, Guarda, Coimbra, Leiria, Viseu, Aveiro]
...	...

Desempenho

Largura Primeiro





Desempenho: Largura

Tempo

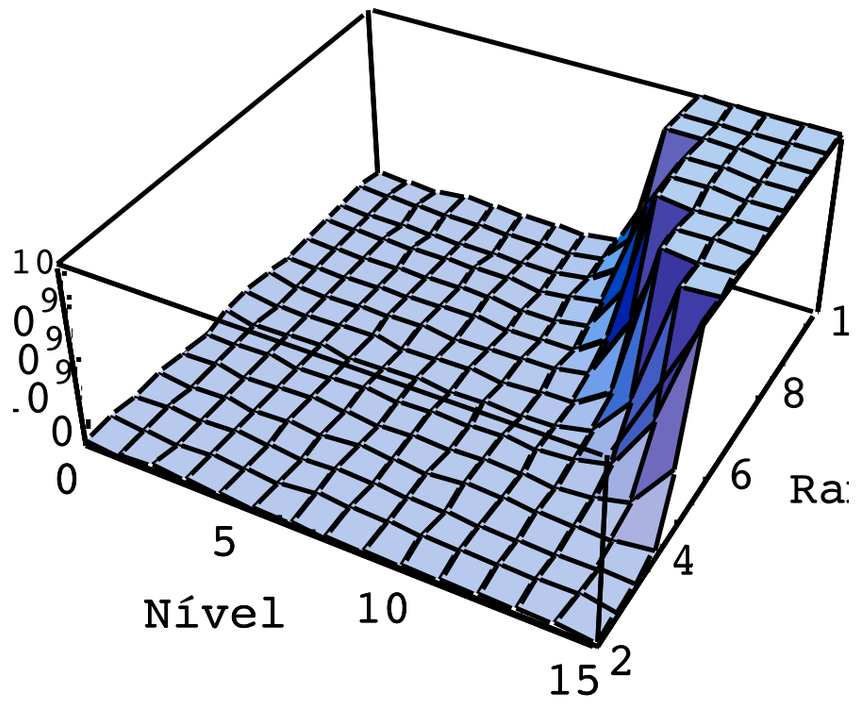
Complexidade Exponencial

$$\sum_{k=0}^{n} r^k = \frac{r^{n+1} - 1}{r - 1} \approx r^n = O(r^n)$$

R= factor de ramificação

Espaço

$O(r^n)$



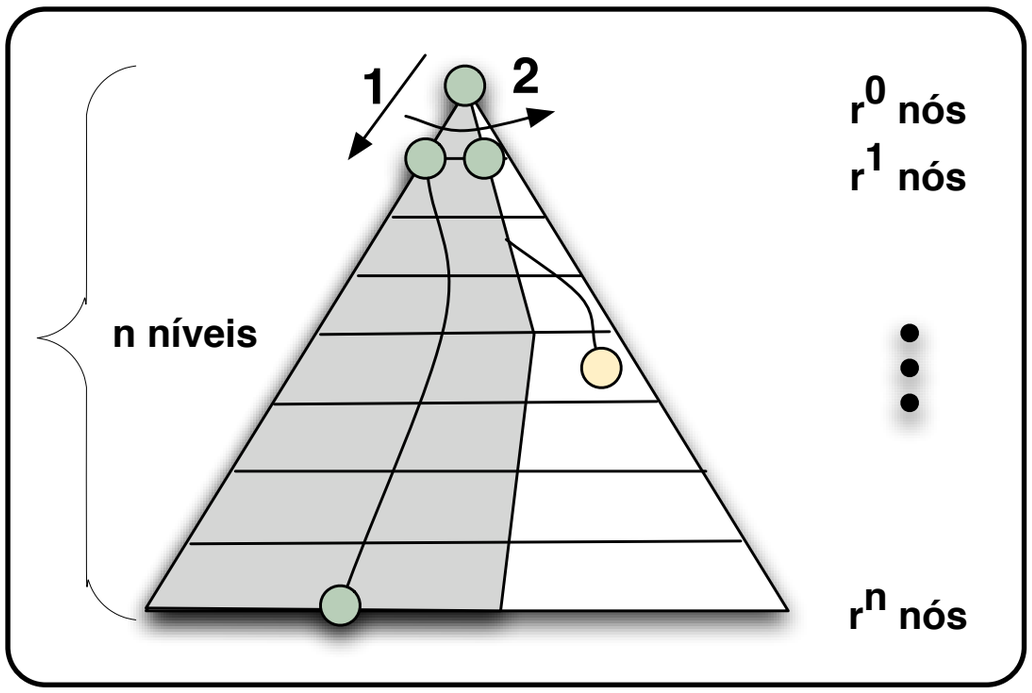
Importância da Complexidade

Considerado:
r=8
1ms por nó
10 bytes por nó

Nível	Nós	Tempo	Espaço
0	1	1ms	10 bytes
5	4681	4,68s	45 Kbytes
10	$153*10^6$	1,9 Dias	1,5 Gbytes
15	$5*10^{12}$	175 Anos	50 Tbytes

Desempenho

Profundidade Primeiro



Desempenho: Profundidade

Tempo

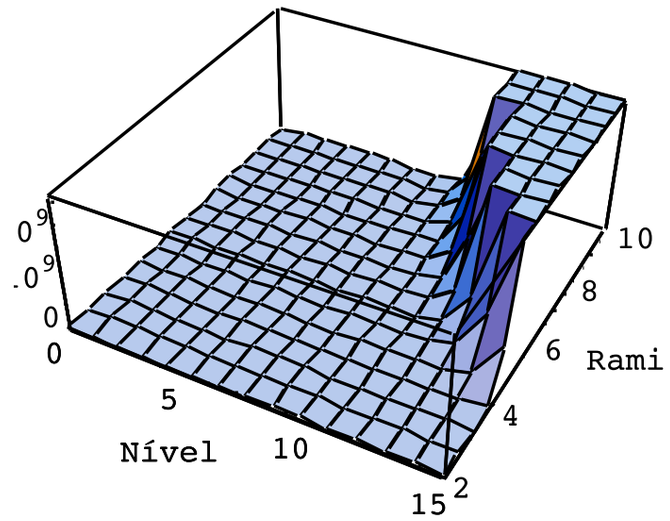
Complexidade Exponencial

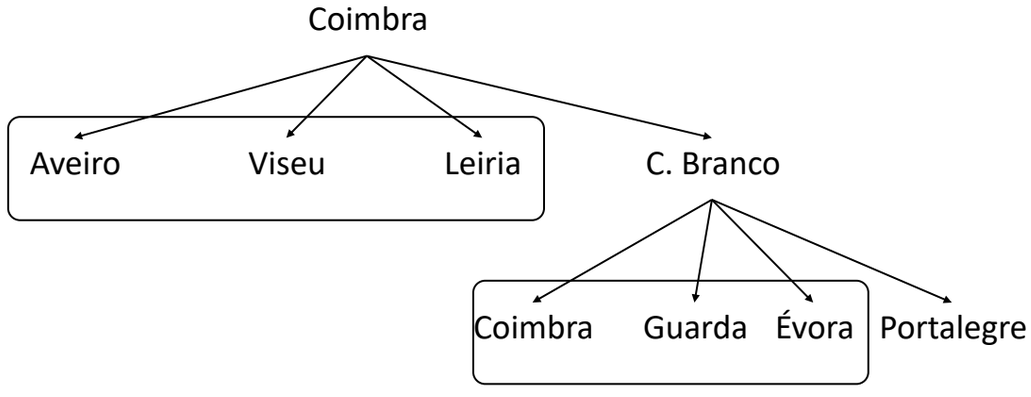
$$\frac{\frac{r^{n+1}-1}{r-1}+(n+1)}{2}=\frac{r^{n+1}+rn-n+r-2}{2(r-1)}\cong O(r^n)$$

Pior — Melhor

Espaço

n(r-1)+1 ≅ O(r*n)





Questões a Considerar

Factor de Ramificação

Profundidade

Ciclos

Outros Algoritmos de Procura Cega

Custo Uniforme

Semelhante a largura primeiro

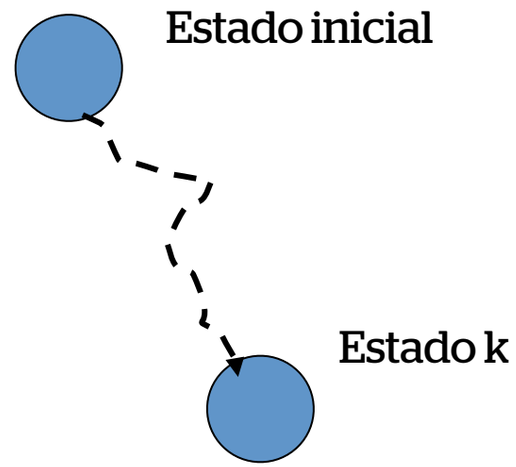
Uso de função de custo g(n).

Completo desde que:

$g(sucessor(k)) \geq g(k)$

Discriminadora

Económica?



g(k) = custo de ir do estado inicial até ao estado K

Custo Uniforme

função ProcuraCustoUniforme (problema, Insere_Ordem_Fila): solução ou falha
1. 1 nós ← Faz_Fila (Estado_Inicial (problema))
2. repete
  2.1 se Vazia_Fila(l_nós)
    então 2.1.1 devolve falha
    fim_de_se
  2.2 nó ← Retira_Fila(l_nós)
  2.3 se Teste_Objectivo(nó)
    então 2.3.1 devolve nó
    senão 2.3.2 Insere_Ordem_Fila(l_nós, Expansão(nó, Operadores(problema)))
    fim_de_se
fim_de_repete
fim_de_função

Iteração	l_nós
0	[[Coimbra, 0]]
1	[[Leiria, 67], [Aveiro, 68],[Viseu, 96], [C. Branco, 159]]
2	[[Aveiro, 68], [Viseu, 96],[Coimbra, 134],[Santarém, 137], [C.Branco, 159], [Aveiro, 182], [Lisboa, 196]]
3	[[Viseu, 96], [Coimbra, 134], [Porto, 136], [Coimbra, 136], [Santarém, 137], [C. Brtanco, 159], [Viseu, 163],] [Aveiro, 182], [Leiria, 183], [Lisboa, 196]]

Profundidade Limitada

Objectivo: Evitar ciclos infinitos

Semelhante a Profundidade Primeiro mas com um limite de profundidade

função ProcuraProfundidadeLimitada (problema, Insere_Pilha, nível_max):
solução ou falha
1.1 nós ← Faz_Pilha (Estado_Inicial (problema))
2. repete
2.1 se Vazia_Pilha(l_nós)
então 2.1.1 devolve falha
fim_de_se
2.2 nó ← Retira_Pilha(l_nós)
2.3 se Teste_Objectivo(nó)
então 2.3.1 devolve nó
senão 2.3.2 Insere_Pilha(l_nós, Expansão(nó, Operadores_Nmx(problema)))
fim_de_se
fim_de_repete
fim_de_função

A indicação do nível faz parte do estado.

Profundidade Limitada

Quando se conhece o nível máximo a que a solução se pode encontrar

É completo

Complexidade Temporal
$O(r^l)$

Complexidade Espacial
$O(r * l)$

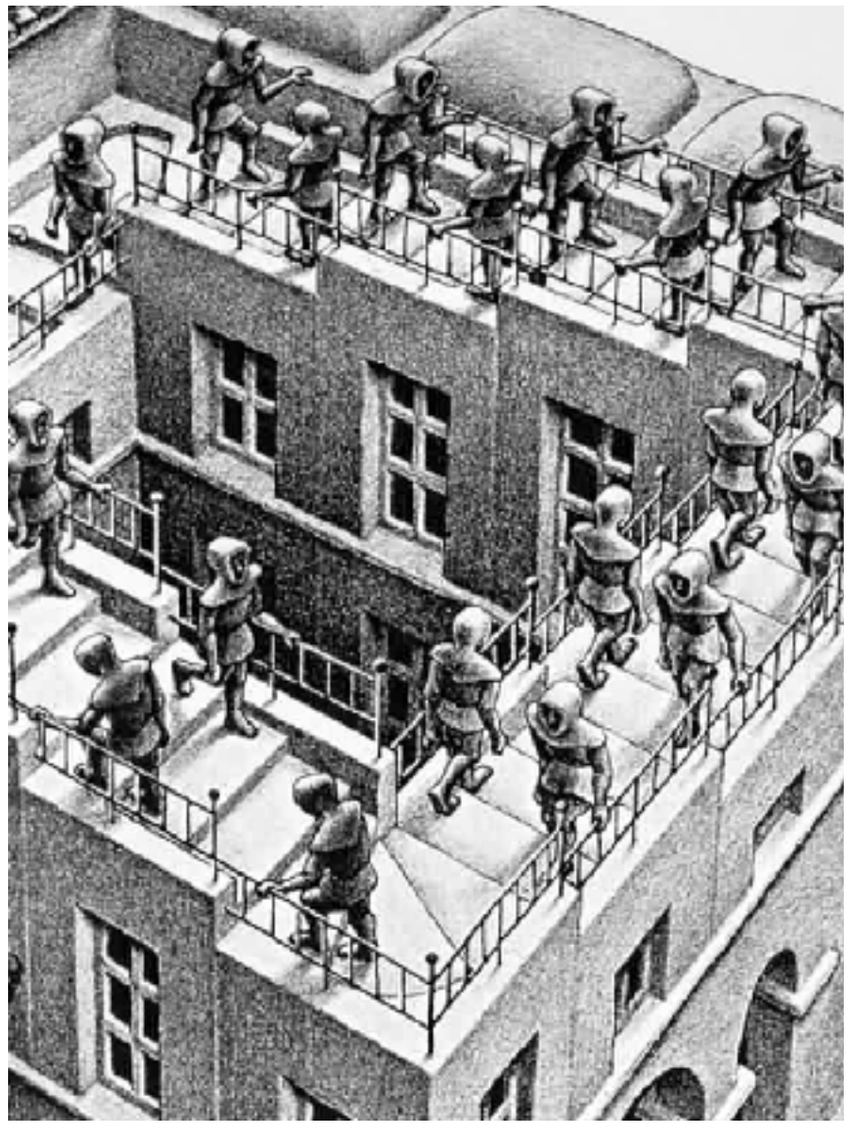
```
função ProcuraProfundidadeLimitada (problema, Insere_Pilha, nível_max):
solução ou falha
  1. l nós ← Faz_Pilha (Estado_Inicial (problema))
  2. repete
    2.1 se Vazia_Pilha(l_nós)
        então 2.1.1 devolve falha
        fim_de_se
    2.2 nó ← Retira_Pilha(l_nós)
    2.3 se Teste_Objectivo(nó)
        então 2.3.1 devolve nó
        senão 2.3.2 Insere_Pilha(l_nós, Expansão(nó, Operadores_Nmx(problema)))
        fim_de_se
fim_de_repete
fim_de_função
```

A indicação do nível faz parte do estado.

Aprofundamento Progressivo

Quando não conhecemos à partida o limite máximo mas queremos resolver o problema dos ciclos e caminhos infinitos...

Iterar o algoritmo de procura limitada



Aprofundamento Progressivo

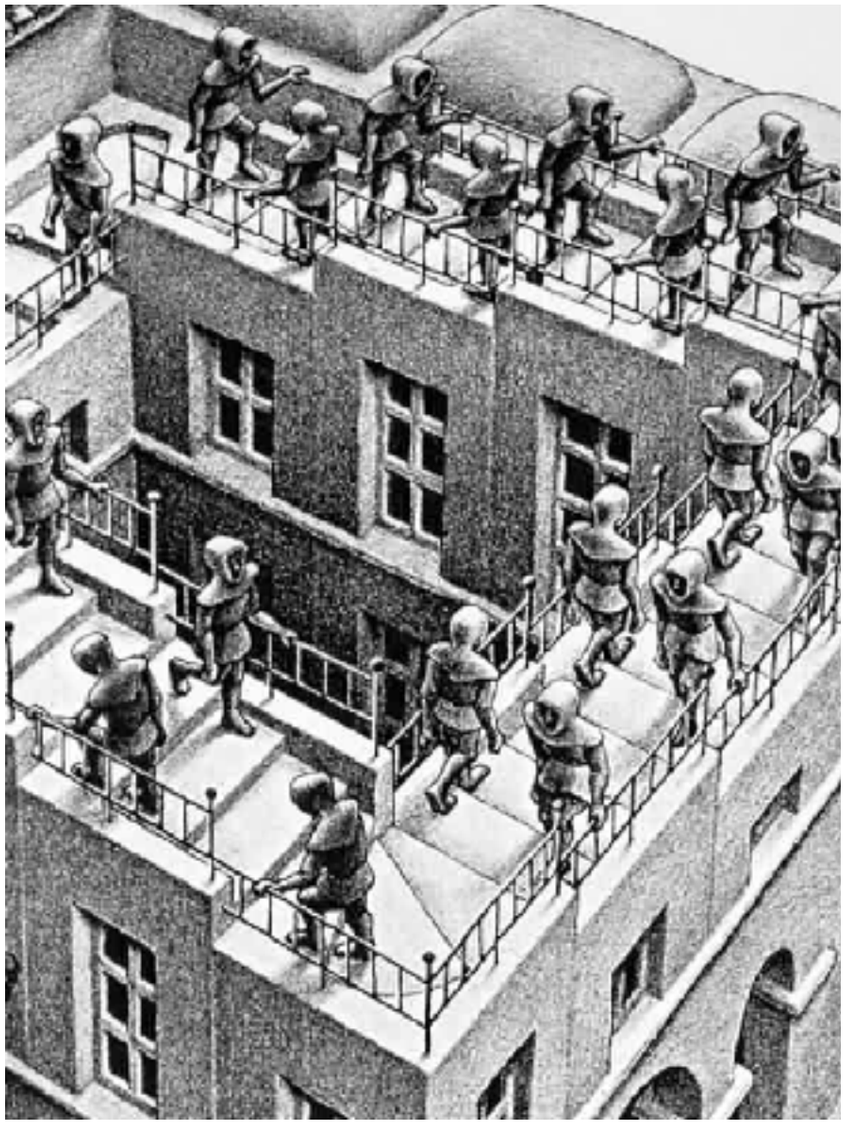
função ProcuraAprofundamentoProgressivo (problema, Insere_Pilha): solução ou falha
1. para nível 0 **até** infinito **faz**
  1.1 se ProcuraProfundidadeLimitada(problema,Insere_Pilha, nível)
então
    devolve solução
    fim_de_se
  fim_de_para
fim_de_função

Aprofundamento Progressivo

Combina aspectos de procura em profundidade com procura por níveis.

Completo?
Sim

Complexidade Espacial
Semelhante à procura em profundidade
O(r*l)
Complexidade Temporal:
Aproximadamente igual à procura em largura
$O(r^n)$



Aprofundamento Progressivo

Se a solução estiver ao nível n

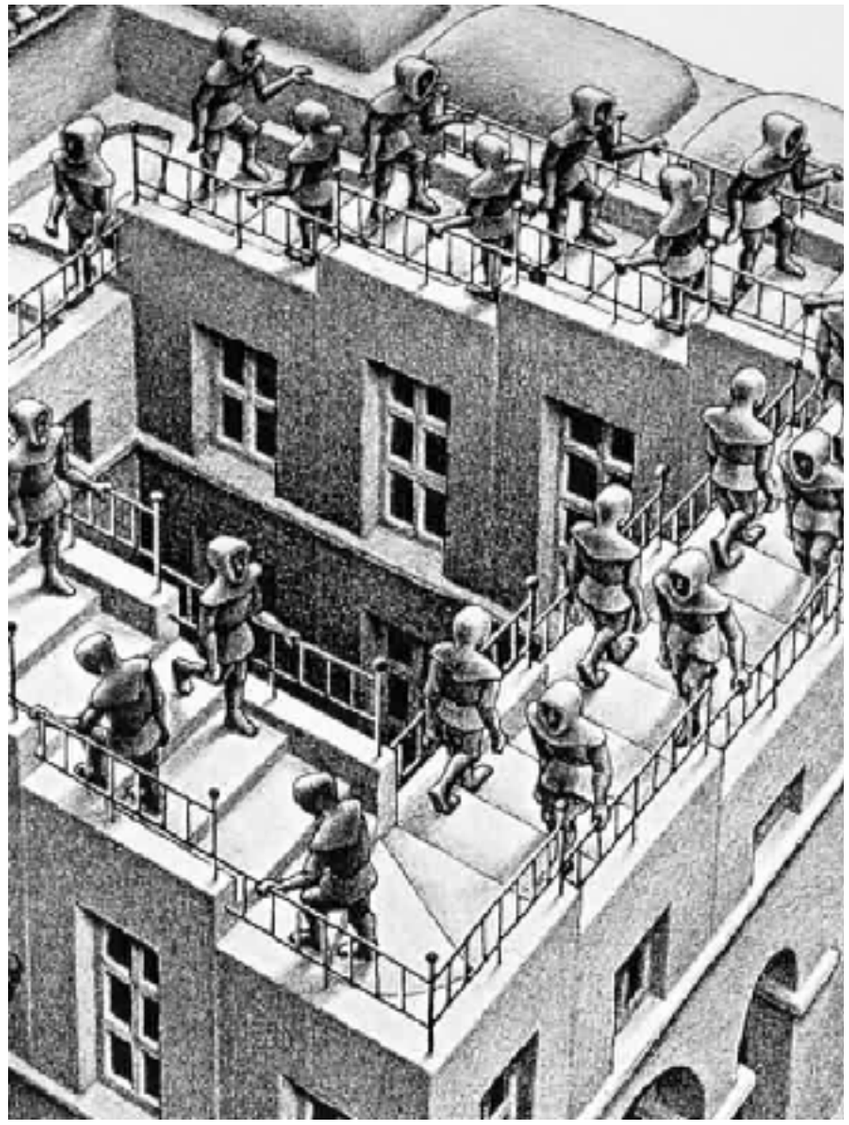
Nós do nível n ➡ 1 vez

Nós do nível n-1 ➡2 vezes

Nós do nível n-i ➡i+1 vezes

Nós do nível 0 (raiz) ➡n+1 vezes

Quanto tempo adicional se perde?



Aprofundamento Progressivo

Usar quando:

O método tem que ser cego

Espaço de procura grande

O nível da solução é desconhecido

$$N_{ap} = \sum_{k=0}^{n} \frac{r^{k+1} - 1}{r - 1} = \frac{1}{r-1}\left[r\left(\sum_{k=0}^{n} r^{k} \right) - \sum_{k=0}^{n} 1 \right] = \frac{1}{r-1}\left[r\left(\frac{r^{n+1} - 1}{r - 1} \right) - (n+1) \right]$$

$$N_{ap} = \frac{r^{n+2} - 2r - rn + n + 1}{(r-1)^{2}}$$

r	n	N_{ap} / N_{lp}
2	5	1.90
	10	1.99
5	5	1.25
	10	1.25
10	5	1.11
	10	1.11

Perde-se aproximadamente:

$$\frac{r}{r-1}$$