

Inteligência Computacional

Rafael D. Ribeiro, M.Sc.
rafaeldiasribeiro@gmail.com
<http://www.rafaeldiasribeiro.com.br>

Inteligência Computacional

Métodos Revogáveis de Busca

- Busca em profundidade
- **Backtracking**
- Busca em largura
- Busca em profundidade iterativa
- Busca ordenada

Inteligência Computacional

Backtracking

- O método de *backtracking* é uma versão especial da busca em profundidade em que apenas um filho é gerado por vez.
- Ele tenta encontrar uma solução gerando um ramo da árvore de busca. Caso este ramo não contenha uma solução, um segundo ramo é gerado. Se este também não possuir uma solução, um outro é gerado e assim o processo continua até que a solução seja encontrada, se esta existir.

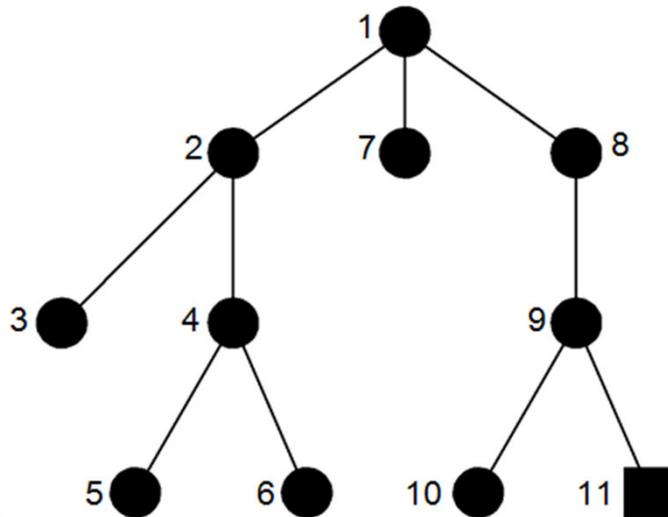
Inteligência Computacional

Backtracking

- A partir do nó raiz um operador selecionado segundo a estratégia definida produz um nó filho.
- Se ele for a **solução** o **processo é interrompido**. Caso **não** seja, um **filho** para este nó também é **produzido e avaliado**, repetindo-se o procedimento de geração e avaliação de nós até que a solução seja encontrada ou quando o método não pode mais continuar.
- Quando isto ocorre, a busca produziu um nó folha. Então, ela retorna um nível na árvore – isto é, retorna ao nó pai do nó folha em questão – e tenta gerar um novo nó filho aplicando outro operador.
- Se obtiver êxito nesta tarefa, a busca prossegue normalmente aplicando as regras segundo a estratégia estipulada. Entretanto, se não for possível gerar algum filho com qualquer uma das regras do problema, a busca retorna mais um nível na árvore e tenta gerar outro filho para o nó pai deste nível do mesmo modo que o anterior.

Inteligência Computacional

Backtracking



Inteligência Computacional

Backtracking

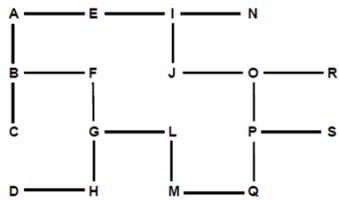
Algoritmo básico de busca Backtracking

```

resposta = nulo
pilha-de-abertos = estado inicial
lista-de-fechados = nulo
sucesso = falso
enquanto (sucesso = falso) e (pilha-de-abertos ≠ vazio) faça
  nó-candidato = elemento do topo da pilha-de-abertos
  se for possível gerar um filho inédito para o nó-candidato
    então
      gere-o
      se nó filho recém-gerado é a solução
        então
          sucesso = verdadeiro
          resposta = nó-filho
        senão
          coloque o nó filho no topo da pilha-de-abertos
      fim-se
    senão
      remova o elemento do topo da pilha-de-abertos
      coloque-o em lista-de-fechados
    fim-se
  fim-enquanto
retorna sucesso e resposta
  
```

Inteligência Computacional

Problema do Labirinto



Método Busca em Backtracking

Entrada: A Saída: S

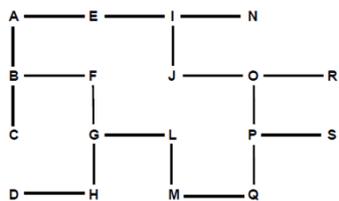
Regras:

- Regra 1: ↓
- Regra 2: ←
- Regra 3: ↑
- Regra 4: →

Estratégia:
As regras são aplicadas na ordem acima

Inteligência Computacional

Problema do Labirinto



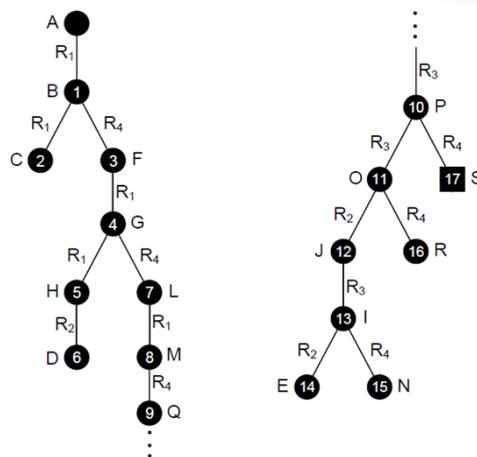
Método Busca em Backtracking

Entrada: A Saída: S

Regras:

- Regra 1: ↓
- Regra 2: ←
- Regra 3: ↑
- Regra 4: →

Estratégia:
As regras são aplicadas na ordem acima



Inteligência Computacional

Problema dos jarros de água

- Existem dois jarros inicialmente vazios. Um possui capacidade igual a 3 litros e o outro igual a 4 litros.
- Ambos podem ser enchidos completamente utilizando uma torneira e podem também ser esvaziados, despejando a água em um ralo.
- Além disso, a água presente em um jarro pode ser passada para o outro.
- Os jarros não possuem marcações e não é permitido o uso de qualquer instrumento de medida. Deseja-se colocar exatamente dois litros de água no jarro maior.
- Pode-se representar as quantidades de água presentes nos dois jarros pelo par ordenado (x,y) , em que x é a quantidade de água no jarro menor e y a quantidade no jarro maior.
- O estado inicial (ambos vazios) é o par $(0,0)$ e o objetivo é encontrar um par do tipo $(x,2)$, isto é, dois litros no jarro maior e qualquer quantidade no jarro menor.

Inteligência Computacional

Problema dos jarros de água

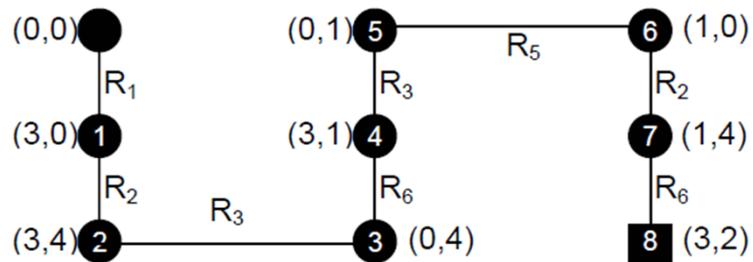
- As regras que definem as ações permitidas são exibidas a seguir, observando que a estratégia de aplicação das mesmas é a ordem de apresentação:

- R1: Se $x < 3$, então $(3,y)$. // regra para encher o jarro menor
 R2: Se $y < 4$, então $(x,4)$. // regra para encher o jarro maior
 R3: Se $x > 0$, então $(0,y)$. // regra para esvaziar o jarro menor
 R4: Se $y > 0$, então $(x,0)$. // regra para esvaziar o jarro maior
 R5: Se $y > 0$ e $x+y \leq 3$, então $(x+y,0)$. // regra para passar toda a água do jarro maior para o menor
 R6: Se $y > 0$ e $x+y > 3$, então $(3,y - (3-x))$. // regra para passar parte do maior para o menor
 R7: Se $x > 0$ e $x+y \leq 4$, então $(0,x+y)$. // regra para passar toda a água do jarro menor para o maior
 R8: Se $x > 0$ e $x+y > 4$, então $(x - (4-y),4)$. // regra para passar parte do menor para o maior

Inteligência Computacional

Problema dos jarros de água

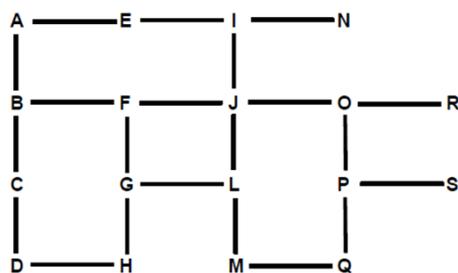
Solução – Método Busca Backtracking:



Em particular, no problema dos jarros de água não houve necessidade de retorno da busca.

Inteligência Computacional

Utilizando o Método de Busca Backtracking, resolva:



Regras

Regra 1: →

Regra 2: ↓

Regra 3: ↑

Regra 4: ←

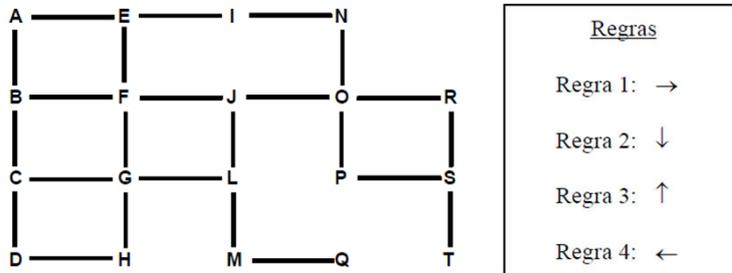
Partida: N

Destino: Q

Estratégia:
As regras são aplicadas na ordem acima

Inteligência Computacional

Utilizando o Método de Busca Backtracking, resolva:



Partida: A
Destino: Q

Estratégia:
As regras são aplicadas na ordem acima

Inteligência Computacional

- Bibliografia utilizada para estas notas de aula:
 - Russel, Stuart J. Inteligência Artificial. Stuart J. Russel. Rio de Janeiro. Elsevier. 2004
 - Notas de Aula do Prof. ROGÉRIO ESPÍNDOLA, disponível na Biblioteca Virtual de docentes – SIA – Estácio, para a disciplina de Inteligência Computacional 1