Inteligência Computacional

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

Inteligência Computacional

Agentes de resolução de problemas

- **Agente:** É um elemento qualquer capaz de perceber seu ambiente por meio de **sensores** e de agir sobre este ambiente por intermédio de **atuadores**.
- **Agentes de resolução de problemas**: Tratam-se de agentes com poder de decisão, ou seja, decidem o que fazer quando encontra **sequência de ações que levam a estados desejáveis**.
  - Devem maximizar sua medida de desempenho.
- **Formulação de objetivos:** é o primeiro passo para a resolução de problemas. Baseada na **situação atual** e na **medida de desempenho do agente**.

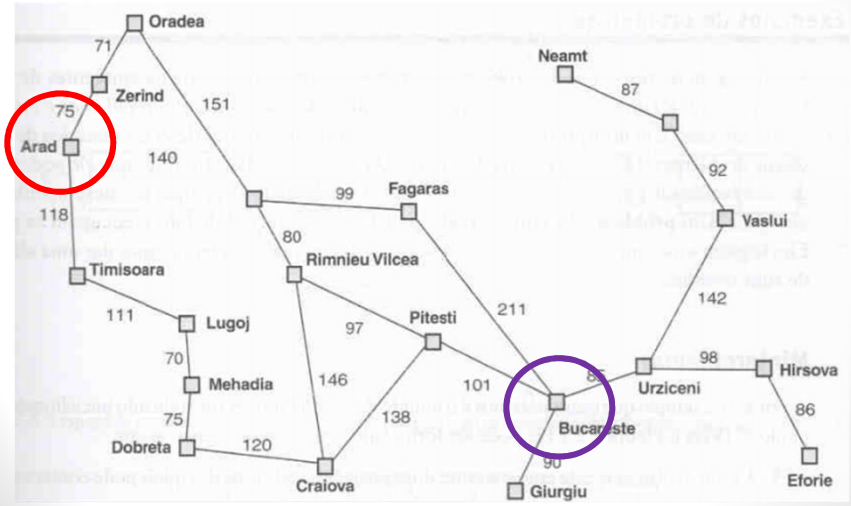
Inteligência Computacional

Agentes de resolução de problemas

- Exemplo: Suponha um agente em férias na cidade de Arad na Romênia. Suas medidas de desempenho são:
  - Melhorar o bronzeado
  - Melhorar seu conhecimento no idioma Romeno
  - Ver as paisagens
  - Apreciar a vida noturna (ver como ela é)
  - Evitar ressacas, etc.
- Suponha ainda que o agente tenha uma passagem aérea não-reembolsável para partir de Bucareste no dia seguinte.
- Faz sentido adotar o **objetivo** de chegar a Bucareste. Os demais cursos podem ser rejeitados sem nenhuma consideração adicional.

Inteligência Computacional

Agentes de resolução de problemas



Inteligência Computacional

FORMULAÇÃO DE OBJETIVOS	Ajudam a organizar o comportamento limitando o comportamento que o agente esta tentando alcançar
FORMULAÇÃO DO PROBLEMA	É o processo de decidir que ações e estados devem ser considerados, dado um objetivo.
BUSCA	Um agente com várias opções imediatas de valor desconhecido pode decidir o que fazer examinando primeiro diferentes sequencias de ações possíveis que levam a estado de valor conhecido, e depois escolhendo a melhor sequencia
SOLUÇÃO	A melhor sequencia

Inteligência Computacional

Agentes de resolução de problemas

- Um algoritmo de busca recebe um problema como entrada e retorna uma solução sob a forma de uma sequência de ações.
- As ações recomendadas podem ser executadas (fase de execução).

Forma-se então o ciclo:

Formular – Buscar – Executar

Inteligência Computacional

Agentes de resolução de problemas

```
função AGENTE-DE RESOLUÇÃO-DE-PROBLEMAS-SIMPLES(percepção) retorna uma ação
    entradas: percepção, uma percepção
    variáveis estáticas: seq, uma seqüência de ações, inicialmente vazia
            estado, alguma descrição do estado atual do mundo
            objetivo, um objetivo, inicialmente nulo
            problema, uma formulação de problema

estado ← ATUALIZAR-ESTADO(estado, percepção)
se seq está vazia então faça
    objetivo ← FORMULAR-OBJETIVO(estado)
    problema ← FORMULAR-PROBLEMA(estado, objetivo)
    seq ← BUSCA(problema)
ação ← PRIMEIRO(seq)
seq ← RESTO(seq)
retornar ação
```

Inteligência Computacional

Agentes de resolução de problemas

- Agentes de Resolução de problemas ← Agentes de ambiente
  - **Ambiente é estático**: não ocorrem mudanças no ambiente durante o processo de formulação e resolução do problema;
  - **Ambiente é observável**: o estado inicial é conhecido
  - **Ambiente é discreto**: os cursos alternativos de ação podem ser enumerados;
  - **Ambiente determinístico:** As soluções para os problemas são sequências de ações únicas. Isto impossibilita a ocorrência de ações inesperadas. Na teoria de controle, denomina-se sistemas similares como laço de repetição aberto ou sistema de malha aberta.

Inteligência Computacional

Agentes de resolução de problemas

Elementos podem definir um problema:

- **Estado inicial**: Trata-se do estado em que o agente começa. Por exemplo:
  - Em(Arad)

- Uma descrição das **ações possíveis** que estão disponíveis para o agente. A formulação mais comum utiliza uma **função sucessor**.

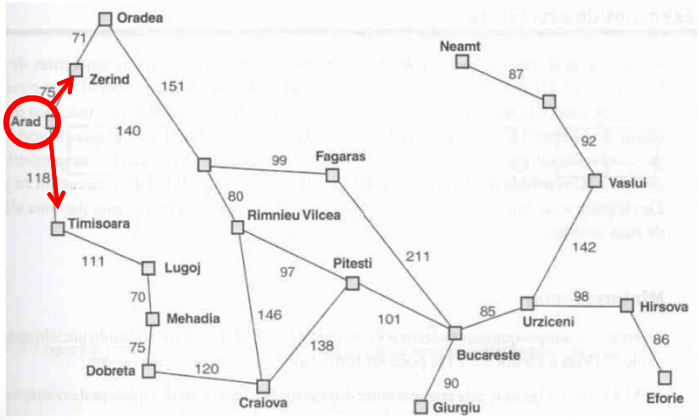
  - Dado um estado particular x, SUCESSOR(x) retorna um conjunto de pares ordenados <ação, sucessor>, em que cada ação é uma ação válida no estado x e cada sucessor é um estado que pode ser alcançado a partir de x aplicando-se a ação.

  - Ex.: Em(Arad) a função sucessor retornaria:
    {<Ir(Timisoara),Em(Timisoara)>,<Ir(Zerind),Em(Zerind)>}

Inteligência Computacional

Agentes de resolução de problemas

- Ex.: Em(Arad) a função sucessor retornaria:
  {<Ir(Timisoara),Em(Timisoara)>,<Ir(Zerind),Em(Zerind)>}



Inteligência Computacional

Agentes de resolução de problemas

- **Espaço de Estados**
  - É a conjunção do estado inicial e a função sucessor do problema. Trata-se do conjunto de todos os estados acessíveis a partir do estado inicial.
  - Forma um grafo em que os nós são estados e os arcos entre os nós são ações.
- **Caminho**
  - Trata-se de uma sequência de estados conectados por uma sequência de ações.
- **Teste de Objetivo:**
  - É um teste que determina se um dado estado é um estado objetivo.
  - É possível a existência de um conjunto de estados objetivos possíveis e o teste simplesmente verifica se o estado dado é um deles.

Inteligência Computacional

Agentes de resolução de problemas

- **Custo de caminho:**
  - É uma função que atribui um custo (valor) numérico a cada caminho. O agente de resolução de problemas escolhe uma função de custo que reflete sua própria medida de desempenho.
  - No exemplo do agente de Arad, o tempo é essencial e, portanto, considera-se o caminho mais rápido aquele com a menor distância percorrida.
- **Custo de passo:**
  - É o custo por se adotar a ação a para ir do estado x ao estado y, ou seja, denota-se c(x,a,y). Para o caso do agente que se desloca de Arad à Bucareste, trata-se das distância das rotas entre os diversos municípios interligados entre as duas cidades.

Inteligência Computacional

Agentes de resolução de problemas

- **Por Exemplo: "*Aspirador de Pó*"**

Estados: O agente está em uma entre duas posições (Esquerda ou Direita), cada uma das quais pode conter sujeira ou não. Desse modo há 2x 2^2 = 8 estados possíveis

Estado Inicial: Qualquer estado pode ser designado como o estado inicial.

Função Sucessor: Gera os estados válidos que resultam da tentativa de executar as três ações (Esquerda, Direita e Aspirar).

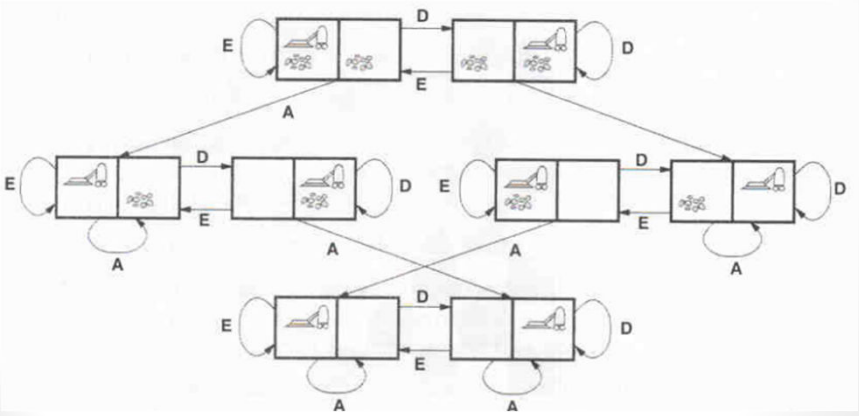
Teste de Objetivo: Verifica se todos os quadrados estão limpos.

Custo de Caminho: Cada passo custa 1, e assim o custo do caminho é o número de passos do caminho.

Inteligência Computacional

Agentes de resolução de problemas

- **Por Exemplo: "*Aspirador de Pó*" - Estados Possíveis**



Inteligência Computacional

- Qualquer tarefa computacional pode ser vista como um problema a ser resolvido. Entretanto, o que aqui é considerado como resolução de problemas é uma tarefa que exige atividades inteligentes, por exemplo, a identificação e a representação do problema e a aplicação de procedimentos que levem a alguma solução.
- Em Inteligência Artificial, grande parte dos métodos de resolução de problemas procuram uma solução dentre um conjunto de possíveis soluções. Esta procura por soluções caracteriza os chamados **problemas de busca**. Os métodos que abordam este tipo de problema são os **métodos de busca**.
- Antes que o processo de busca por soluções seja iniciado, é necessário representar o problema de modo adequado. Para um mesmo problema, diversas representações podem ser definidas.

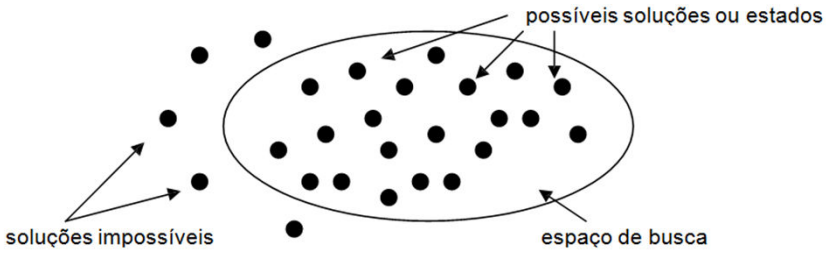
Inteligência Computacional

Sistemas de Produção

- Um sistema de produção é um programa composto por um **conjunto de possíveis soluções**, uma **lista ordenada de regras e um procedimento de controle**.
- As possíveis soluções são chamadas de **estados** e o seu conjunto de **espaço de estados** ou **base de estados** ou **espaço de busca**.
- A representação dos estados leva em consideração todas as variáveis relevantes para o problema. **Uma possível solução não é a resposta para o problema, mas sim uma configuração permitida para as suas variáveis**. Assim, os estados representam as diversas configurações que um problema pode assumir.
- Dois tipos de estados possuem papéis especiais que são o estado inicial e o estado final (uma solução para o problema).

Inteligência Computacional

Sistemas de Produção



Inteligência Computacional

Sistemas de Produção

As regras de um sistema de produção são chamadas de **regras de produção** ou **operadores** e possuem o seguinte formato:

Regra 1: Se condições_1, então ações_1.	R1: c1→a1
...	
Regra i: Se condições_i, então ações_i.	Ri: ci→ai
...	
Regra n: Se condições_n, então ações_n.	Rn: cn→an

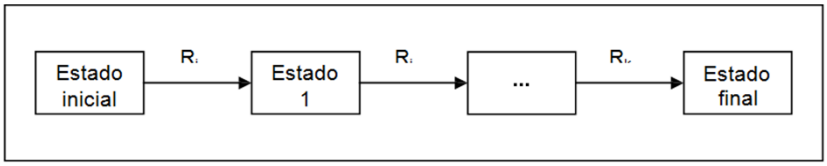
Inteligência Computacional

Sistemas de Produção

- Cada regra representa uma ação que pode ser executada. Para determinar se uma regra será aplicada é necessário que se verifique se o estado atual do problema satisfaz às suas condições.
  - Caso isto ocorra, as ações por ela definidas são efetuadas e o estado atual é transformado em um outro estado, que pode ser novo ou não.
  - Caso as condições não sejam satisfeitas, uma outra regra é avaliada até que alguma ação seja executada.
- Os novos estados obtidos são colocados à disposição do sistema de produção até que seja gerada uma solução final.
- A resolução de um problema constitui-se na aplicação sucessiva de regras que transformam estados em outros até que se encontre um estado final

Inteligência Computacional

Sistemas de Produção



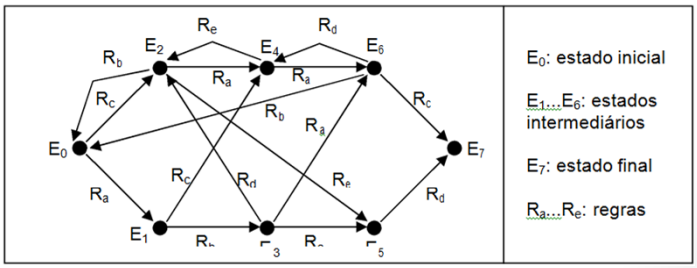
Além de determinar a regra a ser aplicada a cada momento, o **sistema de controle** identifica quando parar o processo. Isto ocorre quando:

- não há suficientes recursos computacionais (por exemplo, pouca memória disponível);
- conclui-se que é impossível ser obtida uma solução para o problema;
- atinge-se um limite arbitrário definido pelo pesquisador (por exemplo, o tempo de processamento);
- encontra-se uma solução satisfatória para um problema.

Inteligência Computacional

Sistemas de Produção

- Uma maneira bastante útil de representar os sistemas de produção é através do **grafo de estados**.
  - Um nó representa um estado e um arco representa a regra cuja aplicação leva um estado a outro
  - Para se resolver um problema deve-se percorrer o seu grafo de estados até que seja obtida uma solução.



Inteligência Computacional

Métodos de busca

De modo simplificado, os métodos ou estratégias de busca podem ser vistos como "receitas" de como realizar as escolhas das ações durante a busca por soluções.

- A medida que um método de busca progride examinando um grafo de estados, um outro grafo de estados é gerado. Chamado de **árvore de busca,** este grafo possui o estado inicial como o nó **raiz**.
- A aplicação dos operadores a um estado produz os seus nós **filhos**, sendo ele chamado de nó **pai**. Observe que o nó raiz é o único que não possui um nó pai. Nós que não possuem filhos são chamados de nós **folhas**. O caminho do nó raiz a um nó folha é chamado de **ramo**.

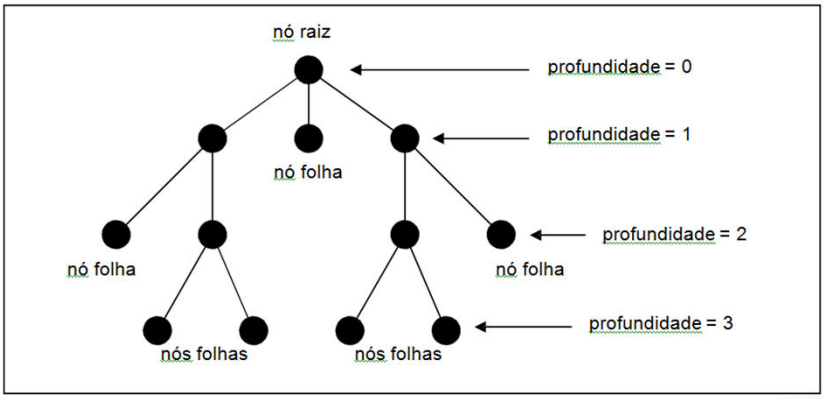
Inteligência Computacional

Métodos de busca

- A **profundidade** de um nó é igual a quantidade de nós existentes do nó raiz, exclusive, até ele. Dessa forma, o nó raiz possui profundidade igual a zero, seus filhos profundidade igual a um, os filhos desses profundidade igual a dois e assim por diante.
- Para alguns métodos de busca a profundidade está diretamente ligada à qualidade de uma solução.

Inteligência Computacional

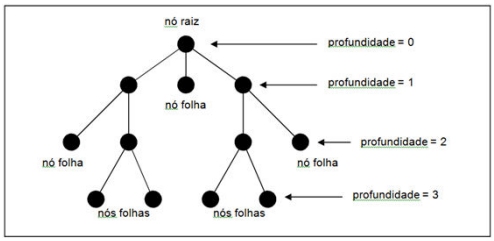
Métodos de busca



Inteligência Computacional

Métodos de busca

- Durante a construção da árvore de busca, o processo de geração dos nós filhos de um determinado nó é chamado de **expansão** deste nó. Assim, um nó pode ser caracterizado como:
  - **a abrir**: um nó que ainda não foi gerado;
  - **aberto** ou **a expandir**: se nenhum de seus nós filhos foi gerado;
  - **explorado**: se algum de seus nós filhos foi gerado;
  - **fechado** ou **expandido**: se todos os seus nós filhos foram gerados.



Inteligência Computacional

Métodos de busca

As estratégias de busca podem ser classificados como:

- **desinformadas** ou **cegas**:
  - não utilizam informações sobre o quão distante um determinado nó está da solução do problema, sendo capaz apenas de identificar se ele é a solução ou não;
- **informadas** ou **heurísticas** ou **guiadas** ou **direcionadas**:
  - utilizam informações sobre o domínio do problema e sobre as soluções desejadas de modo a guiar a busca na direção de regiões mais promissoras do espaço de estados.

Inteligência Computacional

Métodos de busca

As estratégias de busca podem ser classificados como:

- **completas**:
  - se são capazes de explorar todo o espaço de busca;
- **ótimas**:
  - se são capazes de sempre encontrar a melhor solução possível.

Inteligência Computacional

Algoritmo básico para os métodos de busca

```
resposta = nulo
lista-de-abertos = estado inicial
sucesso = falso
enquanto (sucesso = falso) e (lista-de-abertos ≠ vazio) faça
        nó-candidato = algum nó de lista-de-abertos
        se nó-candidato é a solução
         então
                sucesso = verdadeiro
                resposta = nó-candidato
         senão
                expandir o nó-candidato
                coloque o(s) nó(s) gerado(s) em lista-de-abertos
        fim-se
fim-enquanto
retorna sucesso e reposta
```

Inteligência Computacional

Métodos de busca

- Um nó **não** é equivalente a um estado. Além do estado que ele representa, um nó deve armazenar diversas informações importantes para o método de busca tais como a sua profundidade, quem é o seu nó pai, o operador que o gerou e o seu custo quando for necessário

Inteligência Computacional

Busca Desinformada

- As técnicas de busca desinformadas são as mais simples pois não utilizam informações específicas de um problema.
- O método **irrevogável**, que não retorna a situações anteriores.
- Os **revogáveis** que podem retornar a elas caso seja conveniente.

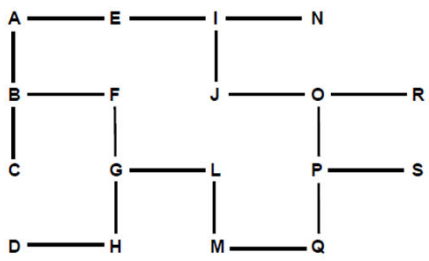
Inteligência Computacional

Algoritmo básico do método irrevogável

```
resposta = nulo
nó-candidato = estado inicial
sucesso = falso
continua =sim
enquanto (sucesso = falso) e (continua = sim) faça
        se nó-candidato é a solução
         então
                sucesso = verdadeiro
                resposta = nó-candidato
         senão
                se existe regra que gere um novo nó-candidato
                 então   nó-candidato = novo nó-candidato
                 senão  continua = não
                fim-se
        fim-se
fim-enquanto
retorna sucesso e reposta
```

Inteligência Computacional

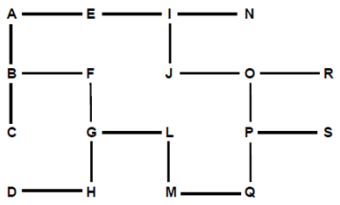
Problema do Labirinto



Entrada: A	Saída: S
Regras:	
Regra 1: ↓	Regra 2: ←
Regra 3: ↑	Regra 4: →
Estratégia:	
As regras são aplicadas na ordem acima	

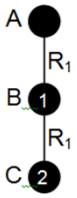
Inteligência Computacional

Problema do Labirinto



Entrada: A	Saída: S
Regras:	
Regra 1: ↓	Regra 2: ←
Regra 3: ↑	Regra 4: →
Estratégia:	
As regras são aplicadas na ordem acima	

Solução – Método Irrevogável:



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

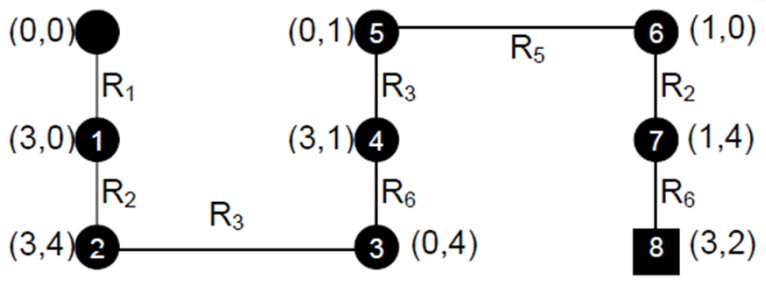
- As regras que definem as ações permitidas são exibidas a seguir, observando que a estratégia de aplicação das mesmas é a ordem de apresentação.

R1: Se $x < 3$, então $(3,y)$. // regra para encher o jarro menor
R_2: Se $y < 4$, então $(x,4)$. // regra para encher o jarro maior
R_3: Se $x > 0$, então $(0,y)$. // regra para esvaziar o jarro menor
R_4: Se $y > 0$, então $(x,0)$. // regra para esvaziar o jarro maior
R_5: Se $y > 0$ e $x+y \leq 3$, então $(x+y,0)$. // regra para passar toda a água do jarro maior para o menor
R_6: Se $y > 0$ e $x+y > 3$, então $(3,y-(3-x))$. // regra para passar parte do maior para o menor
R_7: Se $x > 0$ e $x+y \leq 4$, então $(0,x+y)$. // regra para passar toda a água do jarro menor para o maior
R_8: Se $x > 0$ e $x+y > 4$, então $(x-(4-y),4)$. // regra para passar parte do menor para o maior

Inteligência Computacional

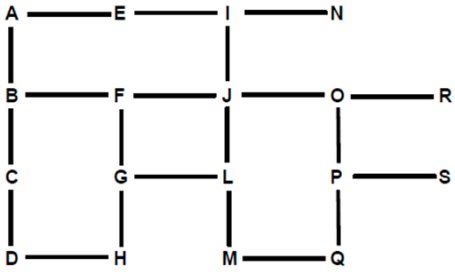
Problema dos jarros de água

Solução – Método Irrevogável:



Inteligência Computacional

Utilizando o Método Irrevogável, resolva:



Regras

Regra 1: →

Regra 2: ↓

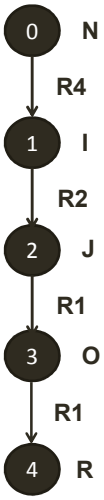
Regra 3: ↑

Regra 4: ←

Partida: N
Destino: Q

Inteligência Computacional

Utilizando o Método Irrevogável, Solução:



Regras

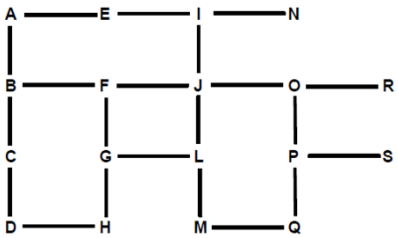
Regra 1: →

Regra 2: ↓

Regra 3: ↑

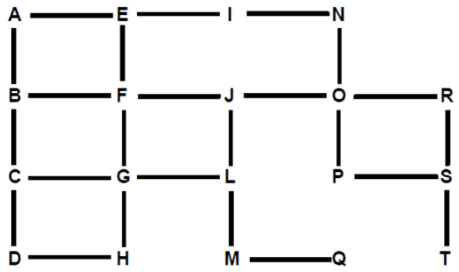
Regra 4: ←

Partida: N
Destino: Q



Inteligência Computacional

Utilizando o Método Irrevogável, resolva:



Regras

Regra 1: →

Regra 2: ↓

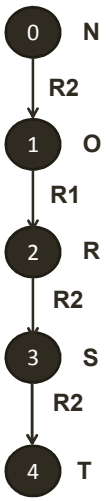
Regra 3: ↑

Regra 4: ←

Partida: N
Destino: Q

Inteligência Computacional

Utilizando o Método Irrevogável, Solução:



Regras

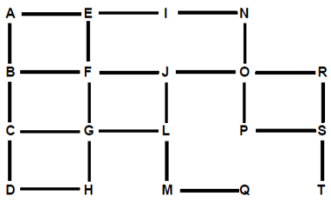
Regra 1: →

Regra 2: ↓

Regra 3: ↑

Regra 4: ←

Partida: N
Destino: Q



Inteligência Computacional

- Bibliografia utilizada para estas notas de aula:
  - Goldschmidt, Ronaldo Ribeiro. Inteligência Computacional / Ronaldo Ribeiro Goldschmidt. Rio de Janeiro: IST-Rio, 2010.
  - 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