

Resolvendo Problemas através de Busca

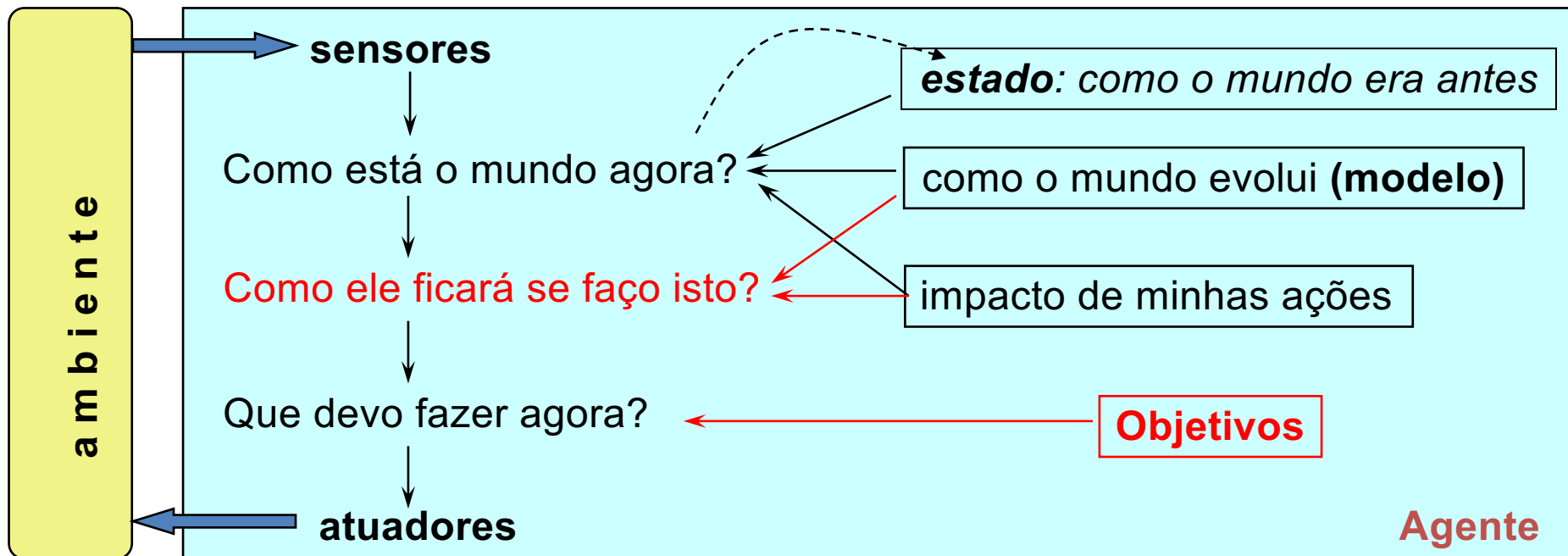
Inteligência Artificial

PCS3438

*Escola Politécnica da USP
Engenharia de Computação (PCS)*

Agente solucionador de problemas (guiado por objetivo – deliberativo)

- Busca uma *sequência de ações* que o leve a estados desejáveis (*objetivos*).



Agente solucionador de problemas

- Propriedades do ambiente para este agente:
 - **Estático:** não muda enquanto o agente delibera (“pensa”)
 - **Discreto:** enumera sequências de estados e alternativas de ações
 - **Determinístico:** solução é uma sequência definida de ações
 - não lida com eventos inesperados ou incertezas
 - executa a sequência definida sem considerar percepções → sistema de controle em malha aberta
 - **Observável:** observa completamente os estados e sabe seu estado inicial
- *Algumas flexibilizações serão feitas em relação às propriedades de **determinismo e observabilidade**.*



Agentes solucionadores de problemas

- O que é um problema em I.A.?
- Como formulá-lo?
- Como buscar a solução do problema?
- Como avaliar a solução e o processo de encontrá-la?

Definição do Problema:

4 componentes

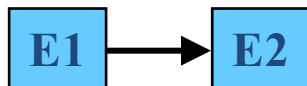
1. **Estado inicial** do problema (onde o agente inicia)
2. Descrição das possíveis **ações** do agente:
 - Pela função sucessor: dado um estado x , **$suc(x)$** retorna um conjunto de pares ordenados **(a,y)** , onde **a** indica cada ação válida em **x** e **y** é o estado sucessor.
 - Pelo conjunto de operadores que podem ser aplicados em um estado para gerar os sucessores.
3. Um teste de **término**:
 - Pode ser um conjunto de estados-objetivos ou
 - Propriedade mais abstrata (ex. cheque-mate em xadrez)
4. Uma **função de custo** da solução
 - avalia numericamente cada solução (medida de desempenho)

Representação de Estados

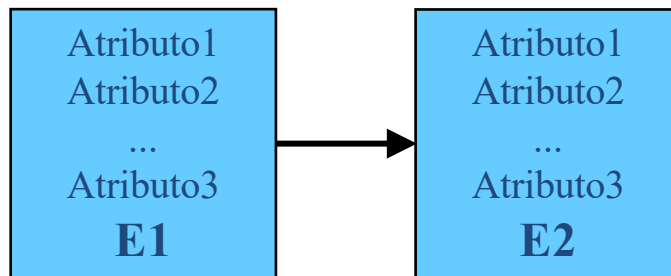
Descrição: Atômica → Fatorada → Estruturada

Complexidade e Expressividade

Atômica:

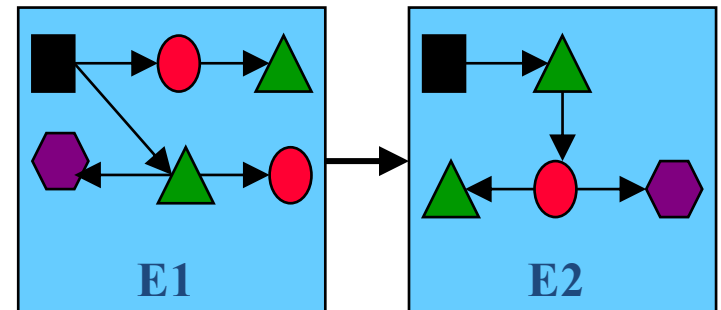


Fatorada:



Vetores com valores de **atributos**
(booleanos, reais, etc)

Estruturada:

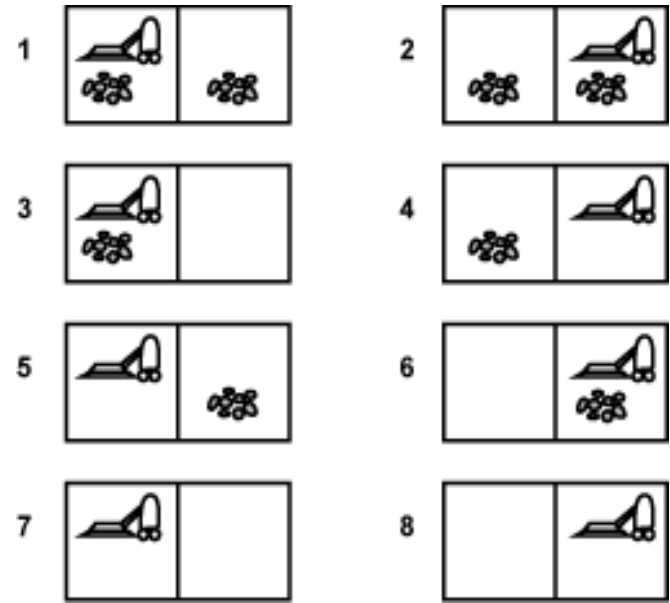


Incluem **objetos**, que podem ter propriedades e **relações** entre eles

Definição de Solução

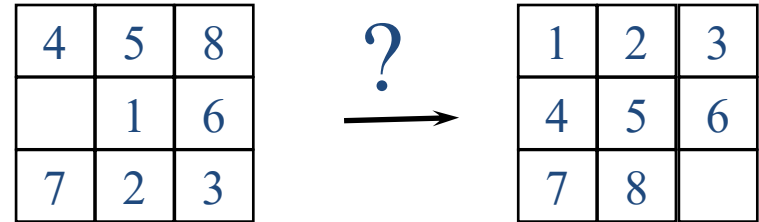
- O estado inicial e a função sucessor implicitamente definem o **espaço de estados** do problema
- O espaço de estados é descrito por um **grafo** onde os vértices representam estados e as arestas, ações.
- Um **caminho** no espaço de estados é uma sequência de estados conectada por uma sequência de ações.
- Uma **solução** para um problema é um caminho do estado inicial para um estado meta (objetivo).
- A **qualidade** da solução é medida pela função de custo da solução

Exemplo 1: Agente Aspirador de Pó



- Formulação do problema:
 - **estado inicial** = qualquer um dos 8 estados acima
 - **função sucessor**: operadores = mover direita (R), mover esquerda (L), aspirar (S); $\text{suc}(1) = \{(R,2), (L,1), (S,5)\}$, etc..
 - **teste de término** = os dois quartos limpos (estado 7 ou 8)
 - **custo do caminho** = quantidade de ações realizadas (ex. custo 1 para cada ação)

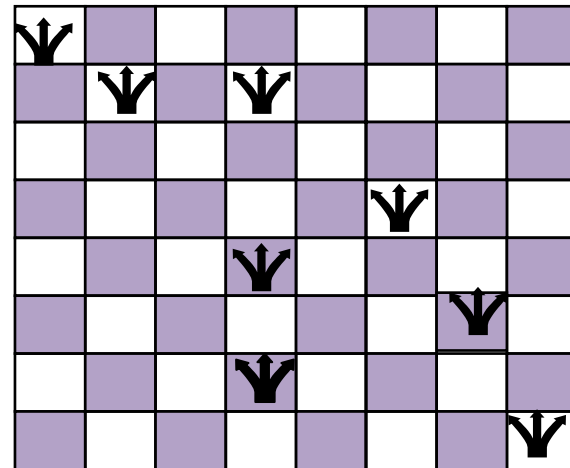
Exemplo 2: Jogo dos 8 números



- **Estado inicial:**
 - cada estado especifica a posição de cada uma das 8 peças e do branco no tabuleiro de 9 posições
 - O estado inicial pode ser qualquer estado
- **Função sucessor:**
 - Gera os estados possíveis que resultam ao aplicar cada uma de 4 ações: branco para esquerda (L), para a direita (R), para cima (U), para baixo (D)
- **Teste de término**
 - Números ordenados, branco em [3,3].
- **Custo do caminho**
 - quantidade de ações realizadas (custo 1 para cada ação)

Importância da formulação

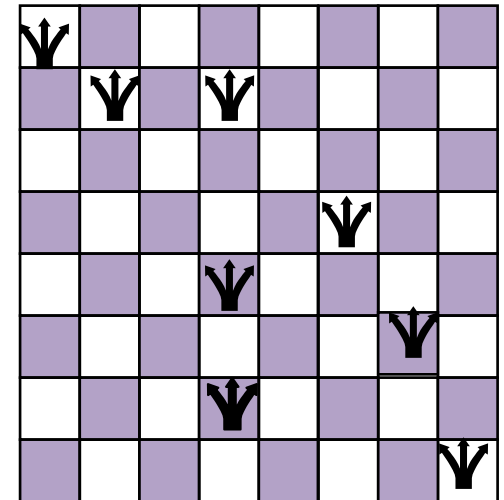
- Jogo das 8 Rainhas
 - **Teste de término:** dispor 8 rainhas de forma que não possam se “atacar”



Importância da formulação

Formulações:

1. Incremental: envolve operadores que “crescem” a descrição de estado, iniciando com um tabuleiro vazio
2. Estado completo: envolve operadores que mantêm a descrição do estado do mesmo “tamanho”, só alterando a configuração.



Importância da formulação

1. Formulação incremental

- **Estado inicial:** tabuleiro sem rainhas; estado: qualquer disposição de 0 a 8 rainhas no tabuleiro
- **Função sucessor:** adicionar uma rainha a qualquer célula vazia
- **Teste de término:** 8 rainhas sem ataque mútuo

Nesta formulação, para um tabuleiro de 8x8 e 8 rainhas, o espaço de estados é:

$$= 64 \times 63 \times \dots \times 57 \approx 3 \times 10^{14}$$

Importância da formulação

2. Estado completo

- **Estado inicial:** qualquer estado com disposição das 8 rainhas, uma em cada coluna
- **Função sucessor:** retorna todos os possíveis estados ao mover uma rainha para outra casa na mesma coluna (cada estado tem $8\text{col} \times 7\text{pos} = 56$ sucessores).

Formulação melhor (espaço de estados bem menor!)

Como encontrar a solução?

Uma vez o problema bem formulado, o estado meta deve ser **buscado** no espaço de estados



Árvore de Busca

A busca é representada em uma **árvore de busca**:

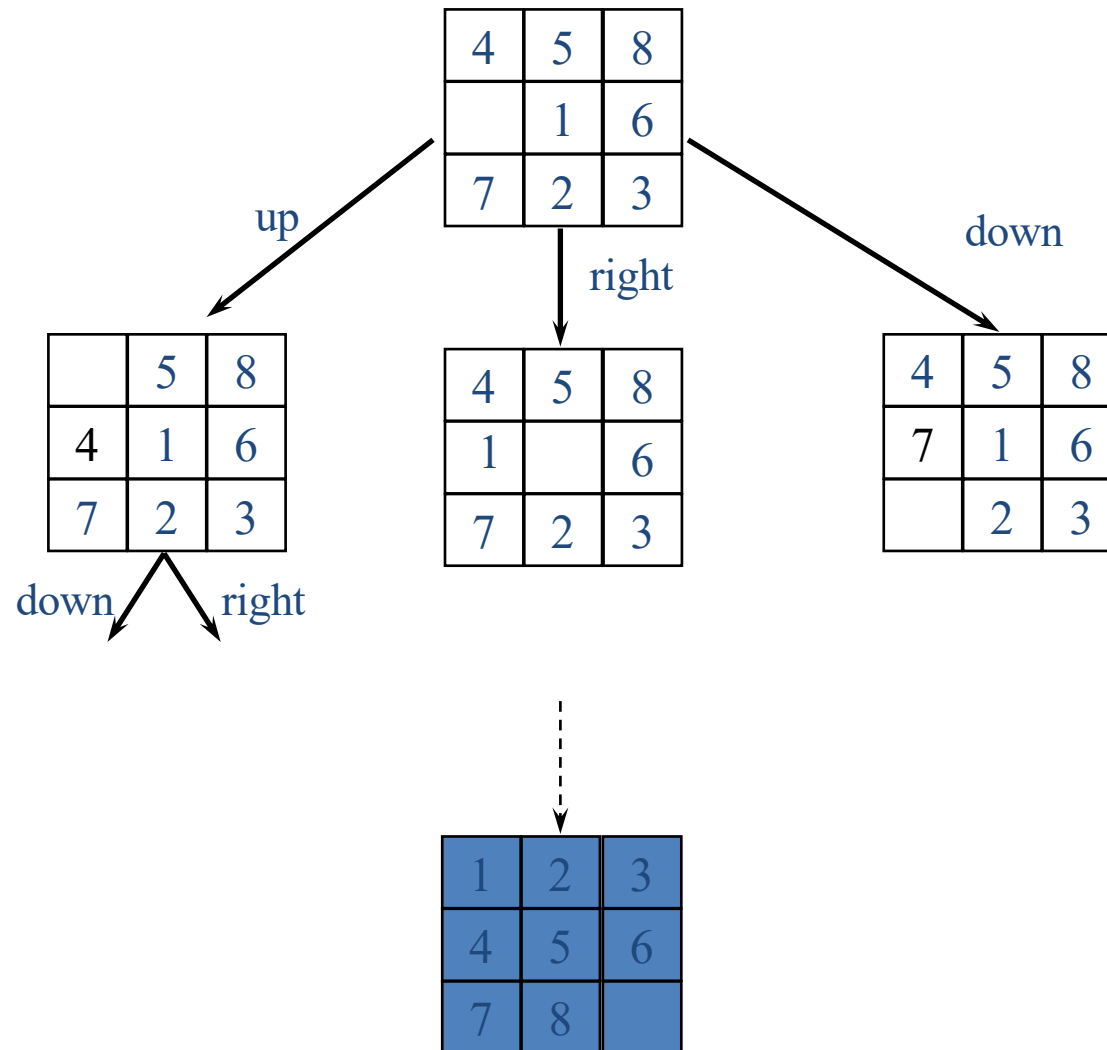
1. **Raiz**: corresponde ao estado inicial
2. **Expande-se o estado corrente** aplicando a função sucessor e gerando um novo conjunto de estados sucessores
3. **Escolhe-se o próximo estado** a expandir seguindo uma **estratégia de busca**
4. **Prossegue-se** até sucesso (atingir um estado meta – retorna solução) ou falha



Espaço de
estados
X
Árvore de
busca

- Espaço de estados $\neq$ árvore de busca!
- Ex.: TSP com 20 cidades
 - espaço de estados = 20 vértices (=cidades)
 - árvore de busca com infinitos vértices (na prática, a busca evita repetir estados)

Árvore de busca para o jogo dos 8 números



Medida de Desempenho na Busca

Desempenho de um algoritmo de busca:

- **Completo:** se existir uma solução, ela certamente é encontrada
- **Ótimo:** a busca encontra a solução de menor custo
- **Complexidade temporal:** quanto tempo demora para encontrar a solução
- **Complexidade espacial:** quanta memória é usada para realizar a busca

Medida de Desempenho na Busca

- Em IA a árvore de busca é tipicamente infinita
- A complexidade é expressa por:
 - **b** – fator de ramificação (*branching*) ou número máximo de sucessores de um nó;
 - **d** – profundidade (*depth*) do nó-meta mais próximo da raiz;
 - **m** – comprimento máximo de um caminho no espaço de estados.

Medida de Desempenho na Busca

- Custo total = custo da solução + custo da busca
 - custo da solução (ex. TSP: caminho a percorrer, em km)
 - custo da busca (tipicamente depende da complexidade em tempo)

Problema: custo da solução (km) + custo da busca (seg)???

- Espaço de estados grande:
 - compromisso (conflito) entre a melhor solução (menor custo da solução) e a solução mais barata (menor custo da busca)



Resumo da aula:

- Problemas podem ser resolvidos por **busca, por um agente guiado por objetivo**, em ambiente totalmente observável, estático, discreto, determinístico
- **Formulação do problema**: estado inicial, ações, teste de término e função de custo
- **Espaço de estados**: grafo com estados como vértices e ações como arestas
- **Solução**: caminho no grafo, do estado inicial a um estado alvo
- A busca por uma solução é representada por uma **árvore de busca**
- Um **algoritmo de busca** pode ser: completo, ótimo e ter suas complexidades temporal e de memória aferidas
- **Medida de desempenho**: custo da solução + custo da busca pela solução