

Roteiro de Aula 1 – Introdução à Inteligência Artificial

Disciplina: BCC740 – Inteligência Artificial

4 de outubro de 2025

1 Objetivos da Aula

- Apresentar o escopo e os objetivos da disciplina.
- Introduzir a noção de **agentes computacionais racionais**.
- Relacionar os principais tópicos do curso segundo Poole & Mackworth (3ª ed.):
 1. Busca em espaços de estados (Cap. 3).
 2. Problemas de satisfação de restrições (Cap. 4).
 3. Sistemas baseados em conhecimento (Cap. 5).
 4. Aprendizado de máquina (Caps. 7 e 8).

2 Visão Geral da Disciplina

O curso segue a estrutura conceitual de **Poole & Mackworth (2023)**, enfatizando a IA como o estudo de *agentes computacionais* que raciocinam e aprendem em ambientes possivelmente incertos e dinâmicos.

Abordagem adotada

- **Modelo de agente:** agente = percepção + raciocínio + ação.
- **Fundamento computacional:** problemas de decisão e otimização.
- **Foco metodológico:** representação, inferência e aprendizado.

Perspectiva unificadora

A IA moderna combina:

- (i) Métodos **simbólicos** (busca, lógica, restrições);
- (ii) Métodos **estatísticos** (aprendizado de máquina);
- (iii) Representações estruturadas e modelos probabilísticos.

3 Agentes Computacionais

Segundo Poole & Mackworth, um **agente computacional** é uma entidade que:

- Recebe **percepções** do ambiente;
- Mantém um **estado interno** que representa o que acredita sobre o mundo;
- Escolhe ações com base em um **modelo computacional de decisão**.

Função agente

$$f : \text{Percepções}^* \rightarrow \text{Ações}$$

Tipos de ambiente

- Determinístico vs. estocástico.
- Estático vs. dinâmico.
- Observável vs. parcialmente observável.
- Discreto vs. contínuo.

PEAS

$$\text{PEAS} = (\text{Performance, Environment, Actuators, Sensors})$$

4 Busca em Espaços de Estados (Capítulo 3)

Definição

Um problema de busca é definido por:

$$\langle S, A, c, s_0, G \rangle$$

onde:

- S : conjunto de estados possíveis.
- A : conjunto de ações aplicáveis.
- $c(s, a, s')$: custo de transição.
- s_0 : estado inicial.
- G : conjunto de estados meta.

Busca não informada

- Busca em largura (BFS).
- Busca em profundidade (DFS).
- Busca de custo uniforme.

```
from collections import deque

def bfs(start, goal, neighbors):
    frontier = deque([start])
    explored = {start: None}
    while frontier:
        state = frontier.popleft()
        if state == goal:
            break
        for nxt in neighbors(state):
            if nxt not in explored:
                explored[nxt] = state
                frontier.append(nxt)
    return explored
```

Busca informada

- Funções heurísticas $h(n)$.
- Algoritmo A*.
- Otimalidade e completude.

5 Problemas de Satisfação de Restrições (Capítulo 4)

Definição

Um problema de satisfação de restrições (CSP) é uma tupla:

$$\langle X, D, C \rangle$$

onde:

- $X = \{X_1, \dots, X_n\}$ são variáveis;
- D_i é o domínio de cada X_i ;
- C é o conjunto de restrições.

Exemplo: coloração de mapa

```
import itertools

cores = ["R", "G", "B"]
vizinhos = {"A": ["B", "C"], "B": ["A", "C"], "C": ["A", "B"]}

for a, b, c in itertools.product(cores, repeat=3):
    if a != b and b != c and a != c:
        print(f"A={a}, B={b}, C={c}")
```

Técnicas de solução

- Busca com retrocesso (*backtracking*).
- Propagação de restrições (AC-3, forward checking).
- Heurísticas de seleção de variáveis (MRV, grau máximo).

6 Sistemas Baseados em Conhecimento (Capítulo 5)

Ideia central

Representar o conhecimento sobre o mundo e inferir novas informações.

Lógica proposicional

$$A \Rightarrow B, \quad A \Rightarrow B$$

Inferência baseada em regras

```
base = [("chuva", "rua_molhada"), ("rua_molhada", "transito_lento")]
fatos = {"chuva"}

for p, q in base:
    if p in fatos:
        fatos.add(q)

print("Inferencias:", fatos)
```

Representação de conhecimento

- Cláusulas definidas: $A \leftarrow B_1, \dots, B_n$
- Encadeamento para frente (data-driven)
- Encadeamento para trás (goal-driven)

7 Aprendizado de Máquina (Capítulos 7 e 8)

Definição

Aprender a partir de dados significa induzir um modelo que generalize:

$$h : \mathcal{X} \rightarrow \mathcal{Y}$$

Tipos de aprendizado

- Supervisionado (classificação, regressão).
- Não supervisionado (agrupamento, redução de dimensionalidade).
- Por reforço (ação e recompensa).

Exemplo: aprendizado supervisionado

```
from sklearn.tree import DecisionTreeClassifier
from sklearn.datasets import load_iris

X, y = load_iris(return_X_y=True)
model = DecisionTreeClassifier().fit(X, y)
print(model.predict([[5.0, 3.5, 1.4, 0.2]]))
```

Componentes de um agente que aprende

- Representação (forma do modelo).
- Avaliação (função de perda, erro).
- Otimização (ajuste dos parâmetros).

8 Resumo e Leituras Recomendadas

- Poole, D. L. & Mackworth, A. K. (2023). *Artificial Intelligence: Foundations of Computational Agents* (3rd ed.). Cambridge University Press.
- Capítulo 2: Agentes e Ambientes.
- Capítulo 3: Busca em Espaços de Estado.
- Capítulo 4: CSPs.
- Capítulo 5: Lógica e Conhecimento.
- Capítulos 7–8: Aprendizado de Máquina.

Atividade sugerida. Descreva um agente computacional para um problema real (ex.: robô aspirador, entrega de encomendas, sistema de recomendação), identificando:

- o ambiente (PEAS);
- o tipo de problema (busca, restrição, conhecimento ou aprendizado);
- e como o agente poderia aprender a melhorar seu desempenho.