UNIVERSIDADE FEDERAL DE MINAS GERAIS
DEPARTAMENTO DE ENGENHARIA ELÉTRICA
FUNDAMENTOS DE INTELIGÊNCIA ARTIFICIAL

Trabalho Prático III

Profs. Cristiano Castro e João Paulo Lara

November 28, 2025

1 Controle de um Robô em um Ambiente Simulado usando Aprendizado por Reforço (Q-Learning)

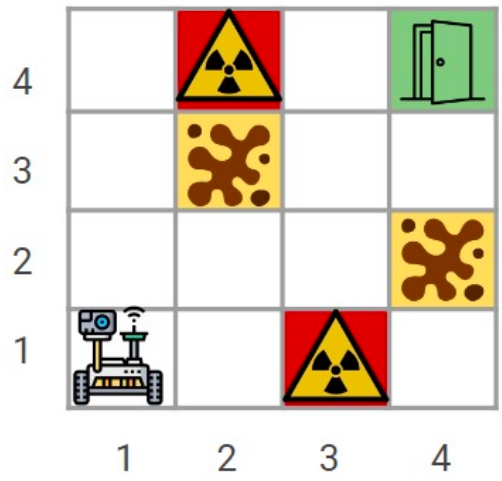

Figure 1.1: Robô no Ambiente Simulado (Grid 4x4).

1.1 Descrição do Problema

Um robô (agente) encontra-se em um ambiente simulando um laboratório de química, com vazamento de algumas substâncias: (i) uma que atola (símbolo de lama) e (ii) uma tóxica que mata (símbolo radioativo). Veja Figura 1.

O agente precisa aprender a navegar nesse laboratório para encontrar a saída com o mínimo de passos possível, sem atolar.

O **espaço de estados** é um grid 4x4, e as **ações** do agente são: ir pra cima, pra direita, pra baixo e pra esquerda, desde que os limites do grid permitam. Por exemplo, na posição $(1, 1)$ ele só pode ir para cima ou para a direita.

Função de Recompensa: cada passo do agente para um estado vazio gera uma recompensa de -1. Se pisar na lama, a recompensa é -5 e, se pisar na substância tóxica, a recompensa é -20 e o episódio termina. Se encontra a saída, a recompensa é +20 e o episódio também termina. Dessa forma, os estados terminais representam os locais onde as substâncias tóxicas se encontram e também a porta de saída do laboratório.

1.2 Objetivo:

Implementar, em *Python*, o algoritmo *Q-Learning* seguindo uma política ϵ-greedy (definição 1.1) para ensinar o agente a navegar nesse laboratório e encontrar a saída com o mínimo de passos possível, sem atolar.

1.3 Dados do Problema:

Esses são os parâmetros iniciais sugeridos para execução do algoritmo *Q-Learning*.

- tamanho do passo (α): 0.2
- desconto (γ): 0.95
- parâmetro do ϵ-greedy (ϵ): 0.1
- número de episódios: 100
- número máximo de iterações por episódio: 10
- o estado de inicialização do agente em cada episódio deve ser escolhido aleatórimente dentre os estados não terminais.

Quando houver mais de uma ação com o maior valor em um dado estado, selecione uma delas aleatoriamente para definir a ação ótima A^* utilizada na política ϵ-greedy.

$$\pi(a \mid S_t) \leftarrow \begin{cases} 1 - \varepsilon + \dfrac{\varepsilon}{|A(S_t)|}, & a = A^*, \\ \dfrac{\varepsilon}{|A(S_t)|}, & a \neq A^*. \end{cases} \tag{1.1}$$

1.4 Entregáveis:

Vocês devem entregar:

- Código-fonte em Python (.ipynb), com comentários claros que explicam a lógica.
- Relatório em PDF (máx. 5 páginas), contendo:
  - Modelagem do Problema.
  - Estratégias adotadas (representação da *Q-table*, escolha da política de seleção de ações - ϵ-greedy, etc.)
  - Gráficos de desempenho (recompensa por episódio, média móvel da recompensa por episódio) variando o parâmetro ϵ
  - A política ótima aprendida, derivada da *Q-table*:
    Pode ser apresentada como um mapeamento estado $\rightarrow$ ação ótima; ou como uma amostragem da política sobre partes do ambiente.
  - Discussão dos resultados.