

# Wumpus

Grupo 02

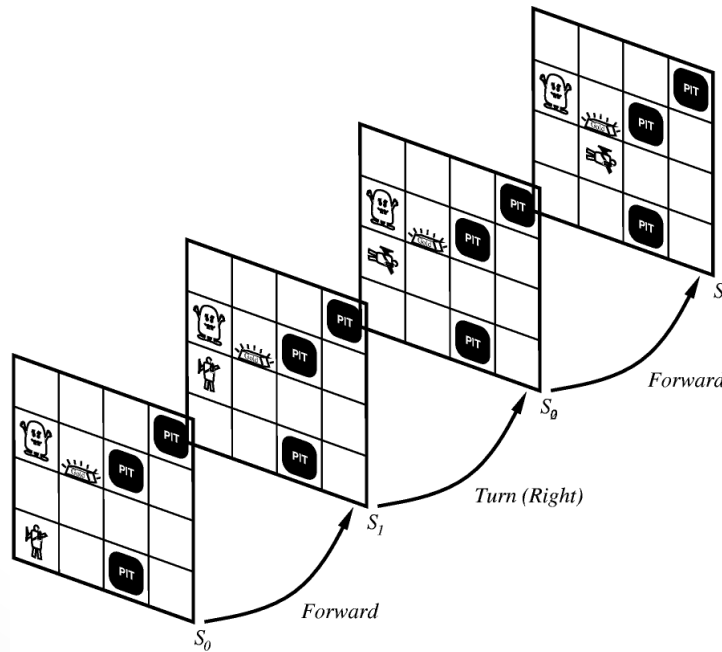
# **Engenharia de Controle e Automação**

Lógica Matemática - Wumpus Grupo 02:

- Hericles Gabriel Escoteiro e Silva;
- Guido de Lyra Ferrario;
- Matheus Pontes Agostinho;
- Jorge Edson Ribeiro da Silva Neto.

## Objetivo do trabalho

- O trabalho tem como objetivo a solução segura para o programa wumpus, que consiste em um tabuleiro repleto de buracos, paredes e, é claro, o monstro Wumpus. Com apenas uma flecha e a percepção de possíveis perigos o agente deve seguir pelo tabuleiro buscando o caminho mais seguro até o ouro, e claro voltar pelo mesmo caminho.



# Problema

- No programa wumpus é definido um tabuleiro de tamanho 4x4 com três buracos, um wumpus e um ouro distribuídos em posições aleatórias;
- O programa sempre é iniciado com o agente na casa [1,1], com apenas uma flecha que pode ser disparada apenas em uma direção e não pode ser recuperada;
- O agente não pode ver o que está no tabuleiro, ele deve andar e memorizar as casas seguras por onde passou, evitando os obstáculos;
- Nas casas adjacentes(acima, abaixo e aos lados) dos buracos haverá brisa e do wumpus haverá fedor esses indicativos devem ser percebidos pelo agente que deve parar imediatamente e tentar contorná-los, até encontrar o ouro;
- Graças ao número limitado de obstáculos, quanto mais brisas e fedores forem detectados maior será a certeza de buraco ou de wumpus, o tiro da flecha pode então ser guardado apenas para o caso de necessidade extrema;
- O brilho do ouro é perceptível apenas na casa do ouro, nada nas redondezas indicará onde ele realmente está, obrigando o agente a explorar ao máximo até encontrá-lo, e então retornar pelo caminho de casas seguras mais curto possível.

# Programa

- A linguagem de programação que pode ser usada é o Prolog, que possui uma ferramenta de listas que pode ser muito útil para resolver diversos problemas envolvendo o programa, claro que utilizadas da maneira correta;
- As listas seguem uma lógica simples. Um conjunto de elementos onde o primeiro elemento é tido como "cabeça" e os outros elementos são tidos como "corpo", as listas podem ter qualquer tamanho e qualquer elemento pode ser "consultado" (no sentido de poder ser utilizado conforme o programador preferir) apenas pela sua posição no conjunto;
- Através dessa ferramenta é possível estabelecer uma relação entre o que o agente pode ou não encontrar na sua casa a cada movimento, como por exemplo [brisa, brilho, fedor], receber uma resposta a cada casa nova visitada, como [no, yes, no], e assim executar uma ação correspondente, que nesse caso seria "pegar o ouro";
- Como o trabalho consiste em pegar o programa pronto do wumpus e fazer o programa agente002 que o solucione, existe uma ordem específica já estabelecida em listas anteriores, então o exemplo anterior na verdade seria (([no, no, yes, \_, \_], grab).

## **Percepções**

- **Fedor: quando o Wumpus está em uma casa adjacente**
- **Brilho: quando o agente está na mesma casa do ouro**
- **Trombada: quando o agente chega a extremidade do mapa**
- **Vento: quando o agente está em uma casa adjacente a um buraco**
- **Grito: acontece quando o Wumpus é acertado por uma flecha e morre**

## **Ações disponíveis**

- **Goforward:** agente anda para a casa a frente
- **Turnleft:** agente gira no sentido anti-horário
- **Turnright:** agente gira no sentido horário
- **Shoot:** agente dispara a flecha
- **Climb:** agente sai da caverna

# FIM