

# Resolvendo jogo Busy Cop em Prolog

Alexandre Regali Seleglim (UFSCar)

## Resumo

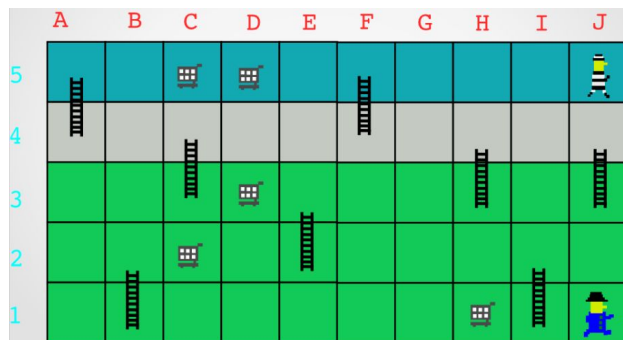
*Usando Prolog, podemos modelar e resolver problemas (neste caso, um problema de estados), onde a solução é encontrada recursivamente usando ações de um agente (inteligência artificial). Neste artigo modelaremos em Prolog uma solução para o jogo Busy Cop.*

**Palavras-Chave:** Prolog; Inteligência Artificial, Problemas de estado

## 1 Introdução

Usando Prolog é possível modelar problemas como problemas de estado, onde a cada decisão tomada por um agente muda o estado atual do problema.

O problema a ser explorado é uma versão simplificada de um jogo chamado Busy Cop (Atari). A versão simplificada adotada neste artigo consiste em uma casa com 5 andares onde cada andar possui 10 espaços horizontais.



Um policial precisa capturar um ladrão dentro dessa casa. Para isso, ele precisará chegar até ele passando por alguns obstáculos. O policial pode se mover livremente pela horizontal, com algumas exceções. Caso haja um carrinho de compras em seu caminho, ele poderá pular por cima dele apenas se não houver nenhum obstáculo do outro lado onde o policial aterrissaria.

## 2 Modelagem

Nossa modelagem começa definindo os objetos (obstáculos) do cenário como fatos em uma base de dados. Exemplos:

```
carrinho(8, 1).
```

```
escada(2, 1).
```

```
paredes(0, 11).
```

```
ladrao([10, 5]).
```

Dessa forma, podemos modificar o cenário do jogo facilmente. Adicionamos e removemos objetos simplesmente adicionando e removendo fatos em nossa base de dados. Os limites do mapa são definidos como constantes de máximo e mínimo que o policial pode ter em suas coordenadas X e Y.

Depois, modelamos nosso estado e uma transição de estado. O estado é a posição atual do policial (uma lista com dois elementos: uma coordenada X e Y). Uma transição é definida como o estado atual (posição do policial) seguida do próximo estado.

```
s([X, Y], [NX, NY])
```

As ações básicas para transitar de um estado para o outro são apenas ações de movimentação (horizontal e vertical). Essas ações passam por verificações para

garantir que o agente se movimenta seguindo as regras do jogo.

### **3 Funções**

Utilizaremos algumas funções para compor nossa solução. Primeiro de tudo, nossa solução será uma lista, contendo uma lista de coordenadas que marcam o caminho percorrido pelo policial até encontrar o ladrão.

Primeiro, executamos uma busca em largura para encontrar a solução do problema. Depois, usaremos funções para estender os caminhos e então juntar todo o caminho pertencente a solução em uma lista. Contaremos com operações básicas em lista, como verificar se um elemento pertence a uma lista e concatenações.

### **4 Conclusão**

Prolog se mostrou uma linguagem poderosa para resolver problemas usando recursão e unificações. Com uma quantidade relativamente pequena de linhas de código conseguimos modelar e resolver um problema como um agente de inteligência artificial.