

Documentação do Trabalho Prático: A Ladra e o Tesouro

Disciplina: CSI107 - Linguagens de Programação

Autor: Bruno Dias Pinto

Matrícula: 18.2.8144

1. Introdução

Este projeto apresenta uma solução em Prolog para o problema de busca de caminho e coleta de itens. O objetivo é modelar uma ladra que deve navegar por um mapa, coletar itens específicos e obter um tesouro final sem ultrapassar o limite máximo de roubos permitido (K). O sistema garante que todas as restrições de conectividade e regras de negócio sejam respeitadas antes de fornecer uma solução.

2. Instruções de Carregamento

O programa foi desenvolvido para ser executado no ambiente **SWI-Prolog** em sistemas Unix/Linux.

Via Terminal:

Bash
swipl ladra.pl

Via Console do Prolog:

Prolog
?- [ladra].

3. Interface de Execução: solve/3

O controle do programa é feito através do predicado principal:

?- solve(Mapa, CidadeInicial, Plano).

Parâmetros:

- **Mapa:** Informe exemplo para carregar o cenário padrão ou custom para utilizar um mapa definido manualmente.
- **CidadeInicial:** O nome da cidade (átomo) onde a jornada começa.

- **Plano:** Variável que receberá a lista de ações (ex: mover, roubar) caso o objetivo seja atingível.

Importante: Ao iniciar a execução, o programa solicitará via teclado que o usuário informe o valor de **K** (capacidade máxima de itens).

4. Modelagem do Estado e Ações

O estado do jogo é representado internamente pela estrutura `estado(CidadeAtual, ItensNaMochila)`. As transições seguem as regras especificadas no enunciado:

- **mover(A, B):** Permite o deslocamento entre cidades desde que exista uma conexão bidirecional.
- **roubar(Item):** Coleta o item exclusivo da cidade atual, respeitando o limite K e evitando duplicatas na mochila.
- **roubarTesouro:** Ação final que exige que a ladra esteja na cidade correta e possua todos os itens requisitos.

5. Validação Automática do Mapa

O sistema realiza uma auditoria completa no mapa antes de iniciar a busca. O `solve/3` falhará (retornando `false`) se as seguintes condições não forem atendidas:

1. **Mínimo de Cidades:** O mapa deve possuir ao menos 4 cidades distintas.
2. **Conectividade Total:** Todas as cidades devem ser alcançáveis a partir do ponto de partida.
3. **Presença de Itens:** Cada cidade do mapa deve possuir ao menos um item exclusivo.
4. **Integridade do Objetivo:** O tesouro deve existir e todos os itens necessários para ele devem estar distribuídos no mapa.

6. Criação de Mapas Customizados

Para testar o programa com um cenário próprio, utilize a seguinte sequência de comandos no prompt:

1. **Limpeza:** `?- limpar_mapa.`

Exemplo:

Prolog

```
?- assertz(conectado(casa, rua)).
?- assertz(conectado(rua, loja)).
?- assertz(conectado(loja, cofre)).
?- assertz(conectado(casa, cofre)).
?- assertz(item_item(casa, lanterna)).
?- assertz(item_item(rua, gazua)).
?- assertz(item_item(loja, luvas)).
?- assertz(item_item(cofre, mapa)).
?- assertz(local_tesouro(cofre)).
```

```
?- assertz(itens_necessarios([gazua, luvas])).  
2.  
3. Resolução: ?- solve(custom, casa, Plano).
```