Inteligência Artificial – Projeto Prolog 2020/2021

Este projeto é composto por dois desafios, a resolução dos mesmo tem de ser feita em Prolog. A entrega do mesmo deve ser composta por dois ficheiros (com extensão *.pl*), um para cada desafio, enviados numa pasta comprimida cujo nome deve possuir o número de aluno e o respetivo nome (exemplo: "12345_JoaoSilva").

A entrega é realizada através do e-learning até 23:59 de 12/04/2021.

Desafio A - Torres de Hanói

A Torre de Hanói é um jogo ou puzzle matemático que consiste em três hastes e vários discos de diferentes tamanhos, que podem ser colocados em qualquer uma das três hastes. O puzzle começa com os discos empilhados por ordem ascendente de tamanho numa haste, o mais pequeno em cima, formando assim uma figura cónica.

O objetivo do puzzle é mover a pilha inteira para a última vara, obedecendo às seguintes regras simples:

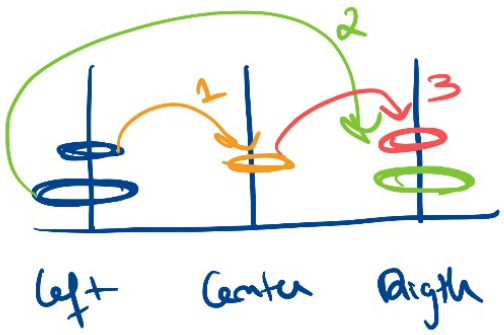
- Apenas um disco pode ser movido de cada vez.
- Cada movimento consiste em retirar o disco superior de uma das pilhas e colocá-lo em cima de outra pilha ou numa haste vazia.
- Nenhum disco de maior dimensão pode ser colocado em cima de um disco de menor dimensão.

Com 3 discos, o puzzle pode ser resolvido em 7 movimentos. O número mínimo de movimentos necessários para resolver um puzzle da Torre de Hanói é 2^n - 1, onde n é o número de discos.

Adaptado de https://en.wikipedia.org/wiki/Tower_of_Hanoi

Define em prolog o predicado move/3, tal que move(N,TA,TD,Taux) significa que se quer mover **N** discos da torre **TA** para a torre **TD** usando a torre **Taux** como auxiliar. O predicado tem de imprimir para a consola (através do predicado write/1) os passos que é necessário realizar.

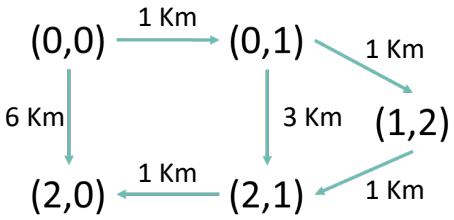
Exemplo:




```
?-  move(2,left,right,center).
Move top disk from left to center
Move top disk from left to right
Move top disk from center to right
```

Desafio B - Caminho entre dois nós

Tendo em conta o grafo direcional da figura, cria os factos edge/3 tal que edge(X,Y,D) significa que existe um arco do nó **X** para o nó **Y** com um comprimento de **D** km. Nota que os nós são representados como pares de coordenadas.



Por exemplo o caminho entre o nó (0,0) e o nó (2,0) deve ser representado por:

```
edge((0,0),(2,0),6).
```

a) Define o predicado path/4, tal que path(X,Y,P,C) siginifica que **P** é um caminho possível entre o nó **X** e o nó **Y** com o comprimento **C**. Entende-se por caminho a lista ordenada de nós entre o nó inicial e o nó final.

Exemplo.

```
?- path((0,0),(2,1),P,C).
P = [(0, 0), (0, 1), (2, 1)],
C = 4;
P = [(0, 0), (0, 1), (1, 2), (2, 1)],
C = 3;
false.
```

Nota: se necessário utilize os predicados reverse/2 e o member/2.

b) Define o predicado shortest_path/4, tal que shortest_path(A,B,P,C) significa que **P** é o caminho mais curto entre o nó **A** e o nó **B** com o comprimento **C.**

Exemplo:

```
?- path((0,0),(2,1),P,C).
P = [(0, 0), (0, 1), (1,2), (2, 1)],
C = 3;
false.
```

Nota: deve usar o predicado findall/3 para obter todas os caminhos possíveis entre os dois nós.