

Trabalho Final



Prof.º: Matheus Nohra Haddad

SIN 323 - Inteligência Artificial

Alunos: Igor Lúcio (igor.lucio@ufv.br)

Gabriel Alves (gabriel.a.ferreira@ufv.br)



Roteiro

1. Definição do Problema
2. Código *Prolog*
3. Resultados

Definição do Problema

- **Ambiente:** tabela 5x5;
- **Movimentação:** $X, Y < 4$ e $X, Y \geq 0$ e apenas quadrantes adjacentes;
- **Caminho:** acíclico;
- **Método de busca:** busca em largura;
- **Parâmetros:** dinâmicos (usuário instância durante execução);
- Máximo 4 pokemons, 1 pokebola, 1 objetivo.

	A	B	C	D	E
1					
2					
3		 3			
4					
5					

REGRAS DE SUCESSÃO

%direita

s([X,Y], [X,Yout]):- Y<4, Yout is Y + 1.

%esquerda

s([X,Y], [X,Yout]):- Y>0, Yout is Y - 1.

%cima

s([X,Y], [Xout,Y]):- X>0, Xout is X - 1.

%baixo

s([X,Y], [Xout,Y]):- X<4, Xout is X + 1.

BUSCA EM LARGURA - PROLOG

```
%solucao por busca em largura (bl) solucao_bl(Inicial,Solucao) :-  
bl([[Inicial]],Solucao).
```

```
%Se o primeiro estado de F for meta, então o retorna com o caminho  
bl([[Estado|Caminho]|_],[Estado|Caminho]) :- meta(Estado).
```

```
%falha ao encontrar a meta, então estende o primeiro estado  
%até seus sucessores e os coloca no final da lista de fronteira  
%bl([Primeiro|Outros], Solucao) :-  
estende(Primeiro,Sucessores),  
concatena(Outros,Sucessores,NovaFronteira),  
bl(NovaFronteira,Solucao).
```

BUSCA EM LARGURA - PROLOG

```
%metodo que faz a extensao do caminho até os nós filhos do estado  
%estende([Estado|Caminho],ListaSucessores, Pkm, Pkb):-  
    bagof([Sucessor,Estado|Caminho], %Lista de objetos  
        ( s(Estado,Sucessor),not(on(Sucessor,[Estado|Caminho])) ),  
        ( conditions(Estado, Sucessor, Pkm, Pkb, Caminho) ),  
        ListaSucessores),!.  
estende(_,[],_,_). %se o estado não tiver sucessor, falha e não procura mais
```

CONDIÇÕES EXTRA

```
conditions(Estado, Sucessor, Pkm, Pkb, Caminho):-  
    s(Estado, Sucessor),  
    on(Pkb, [Estado|Caminho]),  
    not(on(Sucessor, [Estado|Caminho])),  
    on(Sucessor, Pkm).
```

```
conditions(Estado, Sucessor, Pkm, __, Caminho):-  
    s(Estado, Sucessor),  
    not(on(Sucessor, [Estado|Caminho])),  
    not(on(Sucessor, Pkm)).
```



FUNÇÃO-PRINCIPAL

%Funcao principal

```
main(StartingPoint, Pkm, Pkb, Obj, Solution):-  
    search(StartingPoint, Solution, Pkm, Pkb, Obj),  
    write("Pokemons, nos quadrantes: "), writeln(Pkm),  
    write("Pokebola no quadrante: "), writeln(Pkb),  
    howMany(Solution, Pkm, Captured),  
    write("Pokemons capturados: "), writeln(Captured),  
    write("Insignia no quadrante: "), writeln(Obj),  
    write("Caminho do Ash: "),  
    archive(Solution, Pkm, Pkb),!.
```



FUNÇÃO EXTRA - SALVANDO DADOS

```
% Saída do caminho final em arquivo  
archive(Solution, Pkm, Pkb):-  
    open("exit.txt",write, Stream),  
    (writeln(Stream, Solution),  
    writeln(Stream, Pkm),  
    writeln(Stream, Pkb),  
    fail ; true),  
    close(Stream).
```







Tecnologias utilizadas



Código disponível através do endereço:
<https://github.com/iguit0/Inteligencia-Artificial>

FUNÇÃO EXTRA - MOSTRANDO O CAMINHO VISUALMENTE

				
				 3
				