



# Relatório do 5º Trabalho Prático de Inteligência Artificial

Henrique Raposo nº33101

José Serra nº33289

21 de Maio, 2019

## 1 Introdução

Neste trabalho pretende-se construir uma base de Conhecimentos para que um agente possa ter um comportamento inteligente num labirinto. O agente está num labirinto com  $m \times n$  casas. O objectivo é o agente encontrar a saída que está numa casa do labirinto.

## 2 Vocabulário

### 2.1 Fluentes

1. `obstruida(X,Y)`
  - É verdade se a casa  $(X,Y)$  está obstruída
2. `visitada(X,Y)`
  - É verdade se a casa  $(X,Y)$  já foi visitada pelo agente.
3. `esta(X,Y)`
  - É verdade se o agente está na casa  $(X,Y)$
4. `saida(X,Y)`
  - É verdade se a casa  $(X,Y)$  tem a saída do labirinto

### 2.2 Percepções

1. `brisa(X,Y)`
2. `obstruidas(X,Y)`

### 2.3 Ações

1. `mover(X,Y)`
2. `desistir()`
3. `sair()`

## 3 Prolog/Consequências/Lógica

### 3.1 Consequências Pos.

#### 3.1.1 sair()

```
%Prolog sair()
h(ganhou,r(sair,S)):-h(esta(X,Y),S),h(saida(X,Y),r(sair,S)).
```

```
%Logica sair()
h(esta(X,Y),S),saida(X,Y) ==> h(ganhou,r(sair,S))
Se a casa de saida for (X,Y),ou seja, se sentir brisa nessa casa então esta é a
casa de saida
```

#### 3.1.2 mover()

```
%Prolog mover(X,Y)
h(esta(X,Y),r(mover(X,Y),S)):-
    h(esta(K,W),S),
    (K is X+1;W is Y+1;K is X-1; W is Y-1; K is X; W is Y),
    K > 0, K < 7, W > 0, W < 7.
```

```
%Logica mover(X,Y)
h(esta(K,W),S) ==> h(esta(X,Y),r(mover(X,Y),S))
%Vai para a casa (X,Y) se se mover para cima, baixo, direito ou esquerdo.
```

#### 3.1.3 desistir()

```
%Prolog desistir()
h(perdeu,r(desistir,S)):-h(esta(A,B),S).
```

```
%Logica desistir()
h(esta(X,Y),S),saida(X,Y) ==> h(perdeu,r(desistir,S))
Se não tiver jogadas favoráveis então desiste e perde.
```

## 4 Inércia

Descreve o que não muda como consequência de uma ação.

```
h(obstruida(X,Y),r(mover(X,Y),S)):-  
    h(esta(W,Z),S).
```

A casa (X,Y) permanece obstruida, se efetuar a ação mover para qualquer direção em S.

```
h(visitada(X,Y),r(_,S)):-  
    h(esta(X,Y),S),  
    h(visitada(X,Y),S).
```

A casa(X,Y) permanece como visitada em S, independentemente da acção que efectue em S quando está na casa(X,Y) em S.

```
h(ganhou(), r(_,S)):-  
    h(ganhou(),S).
```

O jogador permanece vencedor seja qual for a ação.

```
h(perdeu(), r(_,S)):-  
    h(perdeu(_),S).
```

O jogador permanece vencido seja qual for a ação.