

5º Trabalho de Inteligência Artificial

2018/2019



UNIVERSIDADE
DE ÉVORA

Agente inteligente num labirinto

Docente:
Irene Pimenta Rodrigues

Realizado por:
Daniel Serrano – 35087
Miguel Serrano - 34149

28 de junho de 2019

Introdução:

Neste trabalho, no âmbito da cadeira de Inteligência Artificial iremos implementar um agente que tem um comportamento inteligente dentro de um labirinto com $m \times n$ casas.

1: Vocabulário

Para modelar condições ou fluentes:

- $obstruída(X,Y)$ – é verdade se a casa (X,Y) está obstruída.
- $visitada(X,Y)$ – é verdade se a casa (X,Y) já foi visitada pelo agente
- $esta(X,Y)$ – é verdade se o agente está na casa (X,Y) .
- $saída(X,Y)$ – é verdade se a casa (X,Y) tem a saída do labirinto.

Para modelar ações:

- $mover(X,Y)$
- desistir.
- sair.

Para representar as percepções:

- brisa.
- $obstruídas(n)$, com n o número de casas adjacentes obstruídas..

2:

a), b) e c) As regras, descritas em Português (a), Prolog (b) e Lógica (c) de primeira ordem para modelar:

i. as consequências positivas de cada ação:

mover(X,Y):

Vai para a casa (X,Y1) se movimentar-se para cima em S quando S esta na casa (X,Y) e não estiver obstruida.

Prolog: $h(esta(X,Y1),r(mover(cima),S)):- h(esta(X,Y),S), \quad Y1 \text{ is } Y+1, \quad \text{tabuleiro}(_,TabY),$
 $Y1 \leq TabY, \quad \backslash + h(obstruida(X,Y1),r(mover(cima),S)).$

Lógica: $h(esta(X,Y),S), \text{not } h(obstruida(X,Y1),r(mover(cima),S)) \Rightarrow$
 $h(esta(X,Y+1),r(mover(cima),S))$

Vai para a casa (X1,Y) se movimentar-se para a direita em S quando S esta na casa (X,Y) e não estiver obstruida.

Prolog: $h(esta(X1,Y),r(mover(direita),S)):- h(esta(X,Y),S), \quad X1 \text{ is } X+1, \quad \text{tabuleiro}(TabX,_),$
 $X1 \leq TabX, \quad \backslash + h(obstruida(X1,Y),r(mover(direita),S)).$

Lógica: $h(esta(X,Y),S), \text{not } h(obstruida(X1,Y),r(mover(direita),S)) \Rightarrow$
 $h(esta(X+1,Y),r(mover(direita),S))$

Vai para a casa (X,Y1) se movimentar-se para baixo em S quando S esta na casa (X,Y) e não estiver obstruida.

Prolog: $h(esta(X,Y1),r(mover(baixo),S)):- h(esta(X,Y),S), \quad Y1 \text{ is } Y-1, \quad Y1 > 0, \quad \backslash +$
 $h(obstruida(X,Y1),r(mover(baixo),S)).$

Lógica: $h(esta(X,Y),S), \text{not } h(obstruida(X,Y1),r(mover(baixo),S)) \Rightarrow$
 $h(esta(X,Y-1),r(mover(baixo),S))$

Vai para a casa (X1,Y) se movimentar-se para a esquerda em S quando S esta na casa (X,Y) e não estiver obstruida.

Prolog: $h(\text{esta}(X1,Y), r(\text{mover}(\text{esquerda}), S)) :- h(\text{esta}(X,Y), S), \quad X1 \text{ is } X-1, \quad X1 > 0, \quad \backslash +$
 $h(\text{obstruida}(X1,Y), r(\text{mover}(\text{esquerda}), S)).$

Lógica: $h(\text{esta}(X,Y), S), \text{not } h(\text{obstruida}(X1,Y), r(\text{mover}(\text{esquerda}), S)) \Rightarrow$
 $h(\text{esta}(X+1,Y), r(\text{mover}(\text{esquerda}), S))$

desistir:

Se o agente estiver em S e desistir perde.

Prolog: $h(\text{perdeu}, r(\text{desistir}, S)) :- h(\text{esta}(_X, _Y), S).$

Lógica: $h(\text{esta}(X,Y), S) \Rightarrow h(\text{perdeu}, r(\text{desistir}, S))$

sair:

Se S é a casa de saída e encontra-se na posição (X,Y), e o agente está na posição (X,Y), então esta é a casa de saída e o agente ganha.

Prolog: $h(\text{ganhou}, r(\text{sair}, S)) :- h(\text{esta}(X,Y), S), \text{ saida}(X,Y).$

Lógica: $h(\text{esta}(X,Y), S), \text{ saida}(X,Y) \Rightarrow h(\text{ganhou}, r(\text{sair}, S))$

ii. as leis de inercia, para cada fluente indicar as ações que não alteram o seu valor:

1º:

O agente mantém-se na casa (X,Y), se realizar a acção sair em S e não tem a percepção da brisa em S

Prolog: $h(esta(X,Y),r(sair,S)):- h(esta(X,Y),S), \neg h(brisa,S).$

Lógica: $h(esta(X,Y),S), \text{not } h(brisa,S) \Rightarrow h(esta(X,Y),r(sair,S))$

2º:

O agente mantém-se na casa (X,Y), se realizar a acção mover em S e a casa para a qual se vai movimenter estiver obstruída

Prolog: $h(esta(X,Y),r(mover(Z),S)) :- direcao(Z), \quad h(esta(X,Y),S),$
 $h(obstruida(X,Y),r(mover(Z),S)).$

Lógica: $h(esta(X,Y),S), h(obstruida(X,Y),r(mover(Z),S)) \Rightarrow h(esta(X,Y),r(mover(Z),S))$

3º:

A casa (X,Y) mantém-se obstruída em S, se realizar a acção mover para qualquer lado em S

Prolog: $h(obstruida(X,Y),r(mover(Z),S)):- direcao(Z), \quad h(obstruida(X,Y),S).$

Lógica: $h(obstruida(X,Y),S) \Rightarrow h(obstruida(X,Y),r(mover(Z),S))$

4º:

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

Prolog: $h(visitada(X,Y),S):- h(esta(X,Y),S).$

Lógica: $h(esta(X,Y),S) \Rightarrow h(visitada(X,Y))$

Prolog: $h(visitada(X,Y),r(_S)):- h(visitada(X,Y),S).$

Lógica: $h(visitada(X,Y),S) \Rightarrow h(visitada(X,Y),r(_S))$

5º:

A casa (X,Y) permanece como casa de saída em S, independentemente da acção que se efectue em S

Prolog: $h(saida(X,Y),r(_S)):- h(saida(X,Y),S).$

Lógica: $h(\text{saida}(X,Y),S) \Rightarrow h(\text{saida}(X,Y),r(_,S))$

6º:

Em caso de vitória

Prolog: $h(\text{ganhou},r(_,S)):- h(\text{ganhou},S).$

Lógica: $h(\text{ganhou},S) \Rightarrow h(\text{ganhou},r(_,S))$

7º:

Em caso de derrota

Prolog: $h(\text{perdeu},r(_,S)):- h(\text{perdeu},S).$

Lógica: $h(\text{perdeu},S) \Rightarrow h(\text{perdeu},r(_,S))$

iii. o diagnostico das causas das percepções:

1:

Se há percepção de brisa em S e o agente está em (X,Y), então a casa(X,Y) é a casa de saída.

Prolog: $h(saida(X,Y),S) :- h(brisa,S), h(esta(X,Y),S).$

Lógica: $h(brisa,S), h(esta(X,Y),S) \Rightarrow h(saida(X,Y),S)$

2:

Se há percepção de n casas obstruídas em S, então há n casas próximas que estão obstruídas

Prolog: $proxObst((X,Y),N,Nprox) :- h(esta(X,Y),S), h(obstruidas(N),S), prox((X,Y),Nprox).$

iv. a melhor ação que o agente deve fazer em cada situação:

1. Se o agente está na casa que tem a saída deve sair.
2. O agente deve procurar ir para uma casa que ainda não visitou mas que e segura.
3. Se o agente está rodeado de casas obstruídas e visitadas, desiste.

3: Operadores Tell e Ask

1: Caso em que prefere ganhar

```
ask(kb, accao(A,S)):- h(ganhou,r(A,S)).
```

2: O agente deve procurar ir para uma casa que ainda não visitou e que seja segura

```
ask(kb, accao(mover(cima),S)):- h(esta(X,Y),S),  
    Y1 is Y+1, tabuleiro(_,TabY), Y1=<TabY,  
    \+ h(visitada(X,Y1),_), \+ h(obstruida(X,Y1),r(mover(cima),S)).
```

```
ask(kb, accao(mover(direita),S)):- h(esta(X,Y),S),  
    X1 is X+1, tabuleiro(TabX,_), X1=<TabX,  
    \+ h(visitada(X1,Y),_), \+ h(obstruida(X1,Y),r(mover(cima),S)).
```

```
ask(kb, accao(mover(baixo),S)):- h(esta(X,Y),S),  
    Y1 is Y-1, Y1>0,  
    \+ h(visitada(X,Y1),_), \+ h(obstruida(X,Y1),r(mover(cima),S)).
```

```
ask(kb, accao(mover(esquerda),S)):- h(esta(X,Y),S),  
    X1 is X-1, X1>0,  
    \+ h(visitada(X1,Y),_), \+ h(obstruida(X1,Y),r(mover(cima),S)).
```

3: O agente está rodeado de casas obstruidas e visitadas, desistir

```
ask(kb, accao(disparar(cima),S)):- h(esta(X,Y),S),  
    Y1 is Y+1,  
    obstruida(X,Y1),  
    proxObst((X,Y),N1,Nprox),  
    proxVisitada((X,Y),N2,Nprox),  
    N is N1+N2,  
    N==Nprox.
```

Conclusão:

Com a realização deste trabalho ficámos a ter mais conhecimento em relação ao cálculo de situações, no entanto, achamos que o trabalho ficou a baixo das espetativas pois tem varias falhas.