



UNIVERSIDADE DE ÉVORA

DISCIPLINA DE INTELIGÊNCIA ARTIFICIAL

Base de Conhecimentos - Minas



Autores:

Marcus SANTOS, 29764

Ricardo FUSCO, 29263

Professor:

Irene PIMENTA

RODRIGUES

June 1, 2014

Índice

1	Minas	2
1.1	Vocabulário	2
1.2	Prolog e Lógica de 1 ^a ordem	2

1 Minas

Neste trabalho pretende-se implementar uma base de conhecimentos e um agente inteligente que baseando-se na base de conhecimentos construída até ao momento actual irá movimentar-se de acordo com as percepções que o utilizador dará como input em cada instante. Para ele chegar ao fim basta chegar a uma casa onde sinta brisa e ganha. Este terá as percepções $\text{minas}(X,Y,N)$, nada, ou brisa, e irá mover-se para uma casa de acordo com as percepções que lhe são dadas. Este pode também desistir caso não tenha nenhuma casa segura para onde se mover. Iremos então definir as leis de inércia e as consequências positivas para cada acção.

1.1 Vocabulário

Condições/Fluentes:

- $\text{estar}(X, Y)$
- $\text{segura}(X, Y)$
- $\text{temMina}(X, Y)$
- $\text{visitada}(X, Y)$
- $\text{saida}(X, Y)$

Acções:

- $\text{move}(D)$ - é
- desistir
- sair

Percepções:

- $\text{minas}(X, Y, N)$ - é
- brisa
- nada

1.2 Prolog e Lógica de 1ª ordem

Consequências positivas

Uma acção diz-se ter consequências positivas quando algumas condições/fluentes são alteradas, ou seja, quando alteram o valor de verdade.

Mover:

Prolog: $h(\text{estar}(X,Y),r(\text{move}(P),S)):- h(\text{estar}(K,W),S), \text{val}(X,Y,K,W,P).$

Lógica: $h(\text{estar}(K,W),S) \Rightarrow h(\text{estar}(X,Y),r(\text{move}(P),S))$

Vai para a casa (X,Y) se se mover para P, sendo P cima,baixo, direita e esquerda.

Sair:

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

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

Se a casa de saida for (X,Y), ou seja, se sentir brisa nessa casa então esta é a casa de saída.

Desistir:

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

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

Se a casa de saida for (X,Y), ou seja, se sentir brisa nessa casa então esta é a casa de saída.

Leis da Inércia

Quando uma acção sucede pode haver algumas condições/fluents que se mantêm inalteradas, ou seja, mantêm o valor de verdade.

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

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

Fluente desistiu.

Prolog: $h(\text{visitada}(X,Y),S1):- \text{nonvar}(S1), S1 = r(_,S), h(\text{visitada}(X,Y),S).$

Lógica: $h(\text{visitada}(X,Y),S) \Rightarrow h(\text{visitada}(X,Y),S1)$

A casa (X,Y) permanece como visitada em S1 se já foi visitada num estado anterior em S.

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

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

Se estiver na casa (X,Y) em S e sair o agente permanece na casa (X,Y).

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

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

Fluente ganhou.

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: $h(\text{saida}(X, Y), S) :- h(\text{brisa}, S), h(\text{esta}(X, Y), S).$

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

Este trabalho tem os básicos a funcionar, move-se de acordo com a percepção minas que o utilizador lhe der em cada instante, caso não haja casas seguras para onde mover vai retroceder para as casas que já foram visitadas até chegar a uma onde possa pedir percepção, verifica quais são as casas seguras correctamente, apenas lhe falta o $\text{temMina}(X, Y)$ para verificar se uma casa tem mina de certeza numa determinada situação S. Caso não haja minas dar a percepção $\text{minas}(X, Y, 0)$. Para a percepção $\text{minas}(X, Y, N)$ estão-se apenas a considerar as casas adjacentes. Para correr o programa, basta fazer go.