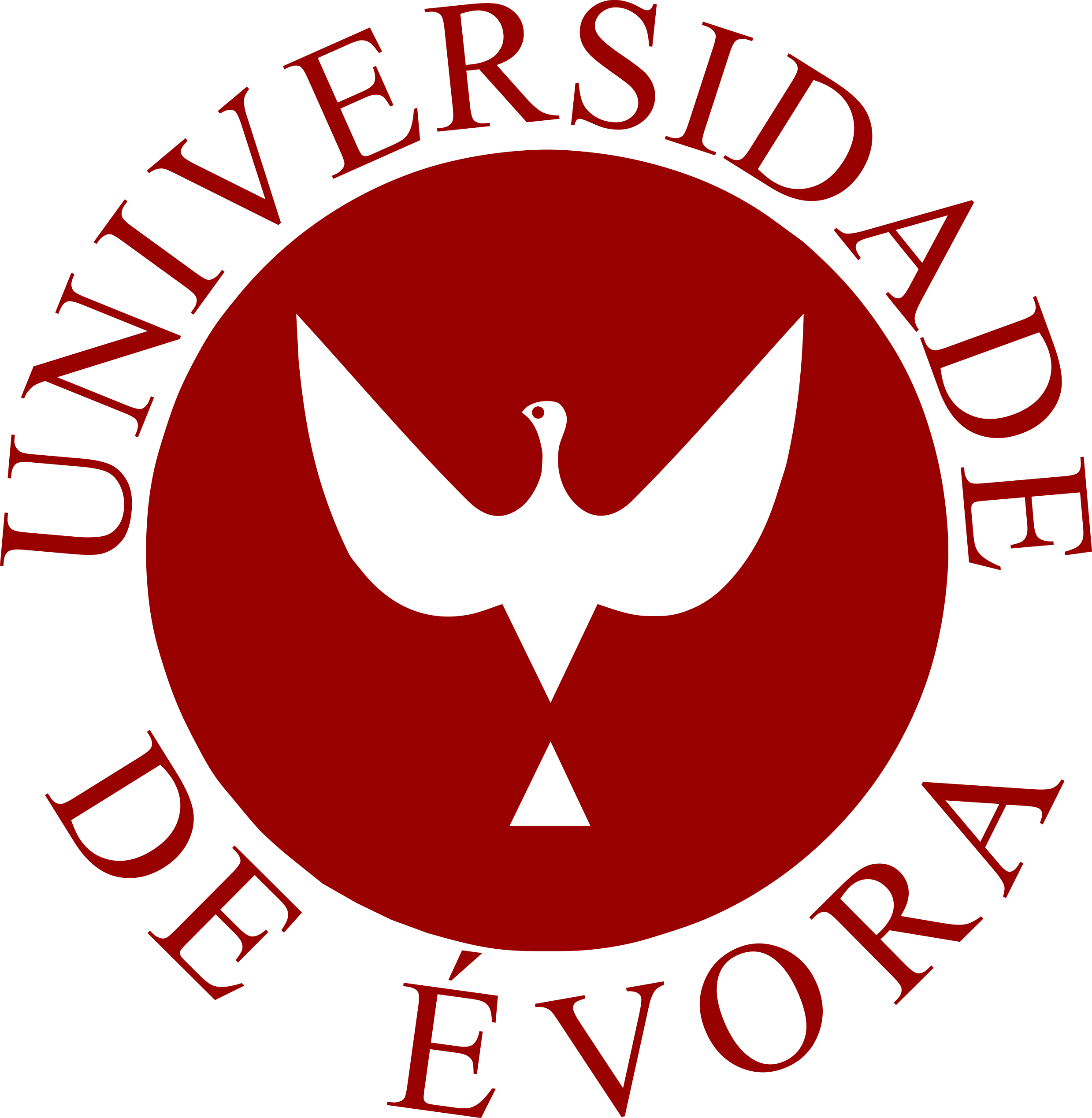
Inteligência Artificial

Trabalho 4

Cálculo de Situações

Discente:

André Gouveia nº 26918

Évora 2015

Indice

[Introdução 2](#_Toc424558633)

[1- Vocabulário 2](#_Toc424558634)

[2- Definição das regras: 3](#_Toc424558635)

[2.1- Consequências positivas das acções 3](#_Toc424558636)

[2.2- Regras que modela lei da inercia 4](#_Toc424558637)

[3- Definição de estado inicial 5](#_Toc424558638)

[4- Query para obter resultado final 5](#_Toc424558639)

[5- Execução 5](#_Toc424558640)

# Introdução

Neste trabalho temos um braço robot em frente a uma sequencia de 4 copos. O robot pode pegar em 2 copos adjacentes e roda-los. Pretende-se definir regras para as acções a tomar pelo braço robot de forma a, dado um estado inicial dos copos, possamos obter um outro estado para os copos.

# Vocabulário

* Condições

pcima(C) - copo virado para cima.

pbaixo(C) - copo virado para baixo.

mao(C1,C2) - copos C1 e C2 na mão.

frente(C1,C2) - está à frente dos copos C1 e C2.

* Acções

viraCC(C1,C2) - vira copos C1 e C2 para cima.

viraBB(C1,C2) - vira copos C1 e C2 para baixo.

viraCB(C1,C2) - vira copo C1 para cima e copo C2 para baixo.

viraBC(C1,C2) - vira copo C1 para baico e copo C2 para cima.

desloca(C1,C2) - passa para a frente dos copos C1 e C2.

agarra(C1,C2) - copos C1 e C2 passam a estar na mão

# Definição das regras:

## 2.1- Consequências positivas das acções

h(pbaixo(C1),r(viraBB(C1,C2),S)):- p(S),

C1\=4,

adjacente(C1,C2),

h(mao(C1,C2),S),

h(pcima(C1),S),

h(pcima(C2),S).

h(pbaixo(C2),r(viraBB(C1,C2),S)):-p(S),

C2\=1,

adjacente(C1,C2),

h(mao(C1,C2),S),

h(pcima(C1),S),

h(pcima(C2),S).

h(pbaixo(C2),r(viraCB(C1,C2),S)):-p(S),

C2\=1,

adjacente(C1,C2),

h(mao(C1,C2),S),

h(pbaixo(C1),S),

h(pcima(C2),S).

h(pbaixo(C1),r(viraBC(C1,C2),S)):-p(S),

C1\=4,

adjacente(C1,C2),

h(mao(C1,C2),S),

h(pcima(C1),S),

h(pbaixo(C2),S).

h(pcima(C1),r(viraCC(C1,C2),S)):-p(S),

C1\=4,

adjacente(C1,C2),

h(mao(C1,C2),S),

h(pbaixo(C1),S),

h(pbaixo(C2),S).

h(pcima(C2),r(viraCC(C1,C2),S)):-p(S),

C2\=1,

adjacente(C1,C2),

h(mao(C1,C2),S),

h(pbaixo(C1),S),

h(pbaixo(C2),S).

h(pcima(C2),r(viraBC(C1,C2),S)):-p(S),

C2\=1,

adjacente(C1,C2),

h(mao(C1,C2),S),

h(pcima(C1),S),

h(pbaixo(C2),S).

h(pcima(C1),r(viraCB(C1,C2),S)):-p(S),

C1\=4,

adjacente(C1,C2),

h(mao(C1,C2),S),

h(pbaixo(C1),S),

h(pcima(C2),S).

h(frente(C1,C2),r(desloca(C1,C2),S)):-p(S),

adjacente(C1,C2),

C1\=4,C2\=1,

h(frente(C3,C4),S),

C3\=C1,C4\=C2.

h(mao(C1,C2),r(agarra(C1,C2),S)):-p(S),

adjacente(C1,C2),

C1\=4,C2\=1,

h(frente(C1,C2),S).

## 2.2- Regras que modela lei da inercia

h(pcima(C1),r(A,S)):-p(S),

C1\=4,

h(pcima(C1),S),

adjacente(C1,C2),

\+ member(A,[viraBB(C1,C2),viraBC(C1,C2)]).

h(pcima(C1),r(A,S)):-p(S),

C1\=1,

h(pcima(C1),S),

adjacente(C2,C1),

\+ member(A,[viraBB(C2,C1),viraCB(C2,C1)]).

h(pbaixo(C1),r(A,S)):-p(S),

C1\=4,

h(pbaixo(C1),S),

adjacente(C1,C2),

\+ member(A,[viraCC(C1,C2),viraCB(C1,C2)]).

h(pbaixo(C1),r(A,S)):-p(S),

C1\=1,

h(pbaixo(C1),S),

adjacente(C2,C1),

\+ member(A,[viraCC(C2,C1),viraBC(C2,C1)]).

h(mao(C1,C2),r(A,S)):-p(S),

C1\=4,C2\=1,

\+ member(A,[viraCC(C1,C2),viraBC(C1,C2),viraBB(C1,C2),viraCB(C1,C2)]),

h(mao(C1,C2),S).

h(frente(C1,C2),r(A,S)):-p(S),

C1\=4,C2\=1,

adjacente(C1,C2),

A \= desloca(1,2),

A \= desloca(2,3),

A \= desloca(3,4),

A \= desloca(4,3),

A \= desloca(3,2),

A \= desloca(2,1),

h(frente(C1,C2),S).

# Definição de estado inicial

adjacente(1,2).

adjacente(2,3). Define as adjacencias entre copos de forma fixa.

adjacente(3,4).

h(pbaixo(1),S0).

h(pcima(2),S0).

h(pbaixo(3),S0). Define os estados iniciais.

h(pbaixo(4),S0).

h(frente(1,2),S0).

p(s0). Define numero finito de passos até encontrar solução.

p(r(\_,s0)).

p(r(\_,r(\_,s0))).

p(r(\_,r(\_,r(\_,s0)))).

p(r(\_,r(\_,r(\_,r(\_,s0))))).

p(r(\_,r(\_,r(\_,r(\_,r(\_,s0)))))).

p(r(\_,r(\_,r(\_,r(\_,r(\_,r(\_,s0))))))).

# Query para obter resultado final

? h(pcima(1),S) , h(pcima(2),S), h(pcima(3),S), h(pbaixo(4),S),h(adjacente(1,2),S), h(adjacente(2,3),S),h(adjacente(3,4),S),h(frente(1,2),S).

# Execução

Para executar o programa basta na pasta do ficheiro trabalho4.pl consultar o mesmo em prolog. Depois fazer as queries da forma indicada no ponto 4 do relatório.