

# Inteligência Artificial

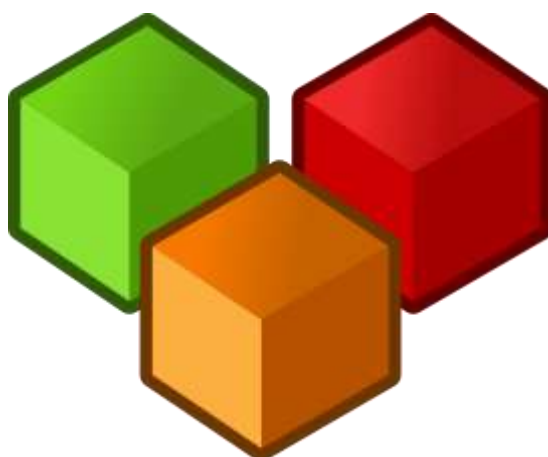
Trabalho 2015/2016

20/06/16



UNIVERSIDADE  
DE ÉVORA

## Planeamento com Cálculo de Situações Trabalho 5



Professora:  
Irene Pimenta Rodrigues

Realizado por:  
João Calhau - 31621  
José Pimenta - 31677

# Índice

|   |        |
|---|--------|
| 0. Introdução.....                              | Pag 3  |
| 1. Notação STRIPS.....                          | Pag 4  |
| 2. Estado inicial em cálculo de situações ..... | Pag 7  |
| 3. Consequências Positivas.....                 | Pag 7  |
| 4. Inércia.....                                 | Pag 9  |
| 5. Query para o plano.....                      | Pag 10 |
| 6. Plano.....                                   | Pag 10 |
| 7. Conclusão.....                               | Pag 11 |

# Introdução

Este trabalho enquadra-se na disciplina de Inteligência Artificial e vamos abordar Planeamento com cálculo de situações para resolução de um exercício definido pela professora, em que temos de resolver um jogo com cubos e ordená-los como pedido, tendo cubos de tamanho 1 e tamanho 2, e temos de retirar cubos para a mão, colocar uns em cima de outros, sobre a mesa, entre outras.

Iremos assim tentar dar o nosso melhor, e iremos tentar utilizar os métodos que achemos mais correctos ou propícios à boa evolução do trabalho e que no final se concretize o que nos é pedido.

Neste trabalho temos um robot com dois braços (1 e 2), e quatro blocos: (A, B, C e D). Os blocos A e D têm o dobro da largura dos blocos B e C. Os blocos com o mesmo tamanho podem ser empilhados uns sobre os outros. Os blocos mais pequenos podem ser empilhados à esquerda ou à direita de um bloco maior. Um bloco grande pode ser empilhado sobre dois pequenos se estiverem lado a lado. O robot pode agarrar num bloco pequeno com um só braço, mas para agarrar num bloco grande precisa dos dois braços.

## 1. Notação STRIPS

(NOTA: livre indica que pode colocar-se peça em cima, chão indica que é a última peça em baixo, bracoDir e bracoEsq são os braços do robô, sobre(X,Y), indica que peça X está sobre Y).

Os estados inicial e final são os seguintes no exercício:

- estado\_inicial([bracoEsq([]), bracoDir([]),  
chao(a),chao(b),  
sobre(d,a), sobre(c,b),  
livre(d), livre(c)]).
- estado\_final([bracoEsq([]), bracoDir([]),  
chao(d), chao(c),  
sobre(a,d), sobre(b,c),  
livre(a), livre(b)]).

**Ex: accao(a1,  
Precondition,  
AddList,  
DeleteList).**

### 1. Desempilha 1 bloco grande que esteja em cima de outro grande e coloca-o nos dois braços

```
acciao(desempilharGrandeDeGrande(D, A),
      [bracoEsq([]), bracoDir([]), livre(D), sobre(D, A)],
      [bracoDir(D), bracoEsq(D), livre(A)],
      [bracoEsq([]), bracoDir([]), livre(D), sobre(D, A)])
:- bloco(D,2), bloco(A,2), A \= D.
```

**2. Desempilha 1 bloco grande que esteja no chão e coloca-o nos dois braços**

```
accao(desempilharGrandeDeChao(A),  
      [bracoEsq([]), bracoDir([]), livre(A), chao(A)],  
      [bracoEsq(A), bracoDir(A)],  
      [bracoEsq([]), bracoDir([]), livre(A), chao(A)])  
:- bloco(A,2).
```

**3. Desempilha 1 bloco pequeno que esteja em cima de outro pequeno e coloca-o em um braço Esquerdo**

```
accao(desempilharPequenoDePequeno(C),  
      [bracoEsq([]), livre(C), sobre(C, B)],  
      [bracoEsq(C), livre(B)],  
      [bracoEsq([]), livre(C), sobre(C, B)])  
:- bloco(B,1), bloco(C,1), B \= C.
```

**4. Desempilha 1 bloco pequeno que esteja no chão e coloca-o em um braço Direito**

```
accao(desempilharPequenoDeChao(B),  
      [bracoDir([]), livre(B), chao(B)],  
      [bracoDir(B)],  
      [bracoDir([]), livre(B), chao(B)])  
:- bloco(B,1).
```

**5. Empilha 1 bloco grande das mãos no chão**

```
accao(empilharGrandeChao(D),  
      [bracoEsq(D), bracoDir(D)],  
      [bracoEsq([]), bracoDir([]), livre(D), chao(D)],  
      [bracoEsq(D), bracoDir(D)])  
:- bloco(D,2).
```

## 6. Empilha 1 bloco pequeno da mão esquerda no chão

```
accao(empilharPequenoChao(C),  
      [bracoEsq(C)],  
      [bracoEsq([]), livre(C), chao(C)],  
      [bracoEsq(C)])  
:- bloco(C,1).
```

## 7. Empilha 1 bloco grande das mãos sobre 1 bloco grande

```
accao(empilharGrandeSobreGrande(A,D),  
      [bracoEsq(A), bracoDir(A)],  
      [bracoEsq([]), bracoDir([]), livre(A)],  
      [bracoEsq(A), bracoDir(A), livre(D)])  
:- bloco(D,2), bloco(A,2), A \= D.
```

## 8. Empilha 1 bloco pequeno das mão direita sobre 1 bloco pequeno

```
accao(empilharPequenoSobrePequeno(B,C),  
      [bracoDir(B)],  
      [bracoDir([]), livre(B)],  
      [bracoDir(B), livre(C)])  
:- bloco(B,1), bloco(C,1), B \= C.
```

## 2. Estado inicial em calculo de situações

h(bracoEsq([]), S0).  
h(bracoDir([]), S0).  
h(chao(a), S0).  
h(chao(b), S0).  
h(sobre(d,a), S0).  
h(sobre(c,b), S0).  
h(livre(d),S0).  
h(livre(c),S0).

## 3. Consequências Positivas

**NOTA: as consequências positivas foram baseadas na notação strips, pensamos assim que não seja necessário explicar novamente o que faz cada ação.**

h(bracoDir(D), bracoEsq(D), livre(A), r(desempilharGrandeDeGrande(D, A), S)) :-  
    h(bracoEsq([]), S),  
    h(bracoDir([]), S),  
    h(sobre(D, A), S),  
    h(livre(D), S).

h(bracoEsq(A), bracoDir(A), r(desempilharGrandeDeChao(A), S)) :-  
    h(bracoEsq([]), S),  
    h(bracoDir([]), S),  
    h(chao(A), S),  
    h(livre(A), S).

h(bracoEsq(C), livre(B), bracoDir(A), r(desempilharPequenoDePequeno(C), S)) :-  
    h(bracoEsq([]), S),  
    h(sobre(C, B), S),  
    h(livre(C), S).

h(bracoDir(B), r(desempilharPequenoDeChao(B), S)) :-  
    h(bracoDir([]), S),  
    h(chao(B), S),  
    h(livre(B), S).

h(bracoEsq([]), bracoDir([]), livre(D), chao(D), r(empilharGrandeChao(D), S)) :-  
    h(bracoEsq(D), S),  
    h(bracoDir(D), S).

h(bracoEsq([]), livre(C), chao(C), r(empilharPequenoChao(C), S)) :-  
    h(bracoEsq(C), S).

h(bracoEsq([]), bracoDir([]), livre(A), r(empilharGrandeSobreGrande(A,D), S)) :-  
    h(bracoEsq(A), S),  
    h(bracoDir(A), S),  
    h(livre(D), S).

h(bracoDir([]), livre(B), r(empilharPequenoSobrePequeno(B,C), S)) :-  
    h(bracoDir(B), S),  
    h(livre(C), S).



## 4. Regras de inércia

**NOTA:** Estas regras representam casos que não podem acontecer, meter algo na mão quando já tem algo, ou tirar algo da mão que não tenha nada.

$h(\text{bracoDir}(B), r(A, S)) :-$   
     $h(\text{bracoDir}(B), S),$   
     $A \backslash= \text{desempilharPequenoDeChao}(B).$

$h(\text{bracoEsq}(B), r(A, S)) :-$   
     $h(\text{bracoEsq}(B), S),$   
     $A \backslash= \text{desempilharPequenoDePequeno}(B).$

$h(\text{bracoDir}(D), \text{bracoEsq}(D), r(B, S)) :-$   
     $h(\text{bracoEsq}(D), S),$   
     $h(\text{bracoDir}(D), S),$   
     $B \backslash= \text{desempilharGrandeDeGrande}(D, A),$   
     $B \backslash= \text{desempilharGrandeDeChao}(A).$

$h(\text{bracoEsq}([], \text{bracoDir}([]), r(X, S)) :-$   
     $h(\text{bracoEsq}([], S),$   
     $h(\text{bracoDir}([], S),$   
     $X \backslash= \text{empilharGrandeSobreGrande}(A, D),$   
     $X \backslash= \text{empilharGrandeChao}(D).$

$h(\text{bracoEsq}([], r(Y, S)) :-$   
     $h(\text{bracoEsq}([], S),$   
     $Y \backslash= \text{empilharPequenoChao}(C).$

$h(\text{bracoDir}([], \text{livre}(B), r(Y, S)) :-$   
     $h(\text{bracoDir}([], S),$   
     $Y \backslash= \text{empilharPequenoSobrePequeno}(B, C).$

## 5. Query

```
:-h(bracoEsq([]),S),  
h(bracoDir([]),S),  
h(chao(d),S),  
h(chao(c),S),  
h(sobre(a,d),S),  
h(sobre(b,c),S),  
h(livre(a),S),  
h(livre(b),S).
```

Representação do estado final .

## 6. Plano obtetido com as definições

```
S =  
r(empilharPequenoSobrePequeno(b,c),  
r(desempilharPequenoDeChao(b),  
r(empilharPequenoChao(c),  
r(desempilharPequenoDePequeno(c),  
r(empilharGrandeSobreGrande(a,d),  
r(desempilharGrandeDeChao(a),  
r(empilharGrandeChao(d),  
r(desempilharGrandeDeGrande(d, a),S0)))))))))
```

# Conclusão

Após a realização deste trabalho, ficámos a conhecer melhor como funcionam algoritmos de planeamento com cálculo de situações. Aplicámos conhecimentos adquiridos de modo a tentar resolver os problemas propostos pela professora, e conseguimos resolver problemas propostos. Criámos ações que vimos necessárias para a resolução do problema, e em grande parte em ajuda pelo trabalho anterior ser um pouco parecido, este correu muito facilmente, ou pelo menos assim nós pensamos.

Em termos de ameaças ou conflitos entre ações, verificámos que existiram ações que têm obrigatoriamente de correr antes de outras, mas que também há certas ações que podem acontecer em várias ordens.