

Inteligência Artificial

Trabalho 2015/2016

17/05/16



UNIVERSIDADE
DE ÉVORA

Planeamento - Cubos



Professora:
Irene Pimenta Rodrigues

Realizado por:
João Calhau - 31621
José Pimenta - 31677

Índice

0. Introdução.....	Pag 3
1. Notação STRIPS.....	Pag 4
2. Estados inicial e final.....	Pag 6
3. POP - Planeador de Ordem Parcial.....	Pag 7
4. Sequência de ações.....	Pag 8
5. Conclusão.....	Pag 9

Introdução

Este trabalho enquadra-se na disciplina de Inteligência Artificial e vamos abordar Planeamento com notação Strips e planeador de ordem parcial para resolução de um exercício definido pela professora, em que temos de resolver um jogo com cubos e ordená-los como pedido.

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).

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

- **accao(desempilharPequenoDePequeno(C),**
 [bracoEsq([]), livre(C), sobre(C, B)],
 [bracoEsq(C), livre(B)],
 [bracoEsq([]), livre(C), sobre(C, B))]
 :- member(B, [b, c]), member(C, [b, c]), B \= C.

- **accao(desempilharPequenoDeChao(B),**
 [bracoDir([]), livre(B), chao(B)],
 [bracoDir(B)],
 [bracoDir([]), livre(B), chao(B))]
 :- member(B, [b, c]).

Estas ações têm algo em comum, que é remover 1 cubo pequeno para a mão. O que diferencia as ações é em 2 casos mover o cubo da posição que está para a mão esquerda ou a mão direita, e outra diferença em 2 casos é o cubo estar em cima de outro cubo pequeno, ou o cubo estar em cima do chão.

- accao(desempilharGrandeDeGrande(D, A),
[bracoEsq([]), bracoDir([]), livre(D), sobre(D, A)],
[bracoDir(D), bracoEsq(D), livre(A)],
[bracoEsq([]), bracoDir([]), livre(D), sobre(D, A)])
:- member(D, [a, d]), member(A, [a, d]), A \= D.

Esta ação retira uma peça grande que esteja em cima de outra peça grande e coloca-a nas mãos do robô.

-----//-----

- accao(empilharGrandeChao(D),
[bracoEsq(D), bracoDir(D)],
[bracoEsq([]), bracoDir([]), livre(D), chao(D)],
[bracoEsq(D), bracoDir(D)])
:- member(D, [a, d]).

Esta ação retira uma peça grande das mãos do robô e coloca-a no chão.

-----//-----

- accao(empilhar2PequenosGrande(B, C, D),
[bracoEsq(C), bracoDir(B), livre(D)],
[bracoEsq([]), bracoDir([]), livre(B), livre(C), sobre(B, D), sobre(C, D)],
[bracoEsq(C), bracoDir(B), livre(D)])
:- member(B, [b, c]), member(C, [b, c]), B \= C, member(D, [a, d]), D \= B,
D \= C.

Esta ação coloca duas peças pequenas da mão do robô para cima de 1 peça grande.

-----//-----

- accao(desempilharGrandeDeChao(A),
[bracoEsq([]), bracoDir([]), livre(A), chao(A)],
[bracoEsq(A), bracoDir(A)],
[bracoEsq([]), bracoDir([]), livre(A), chao(A)])
:- member(A, [a, d]).

Esta ação retira 1 peça grande do chão para as duas mãos do robô.

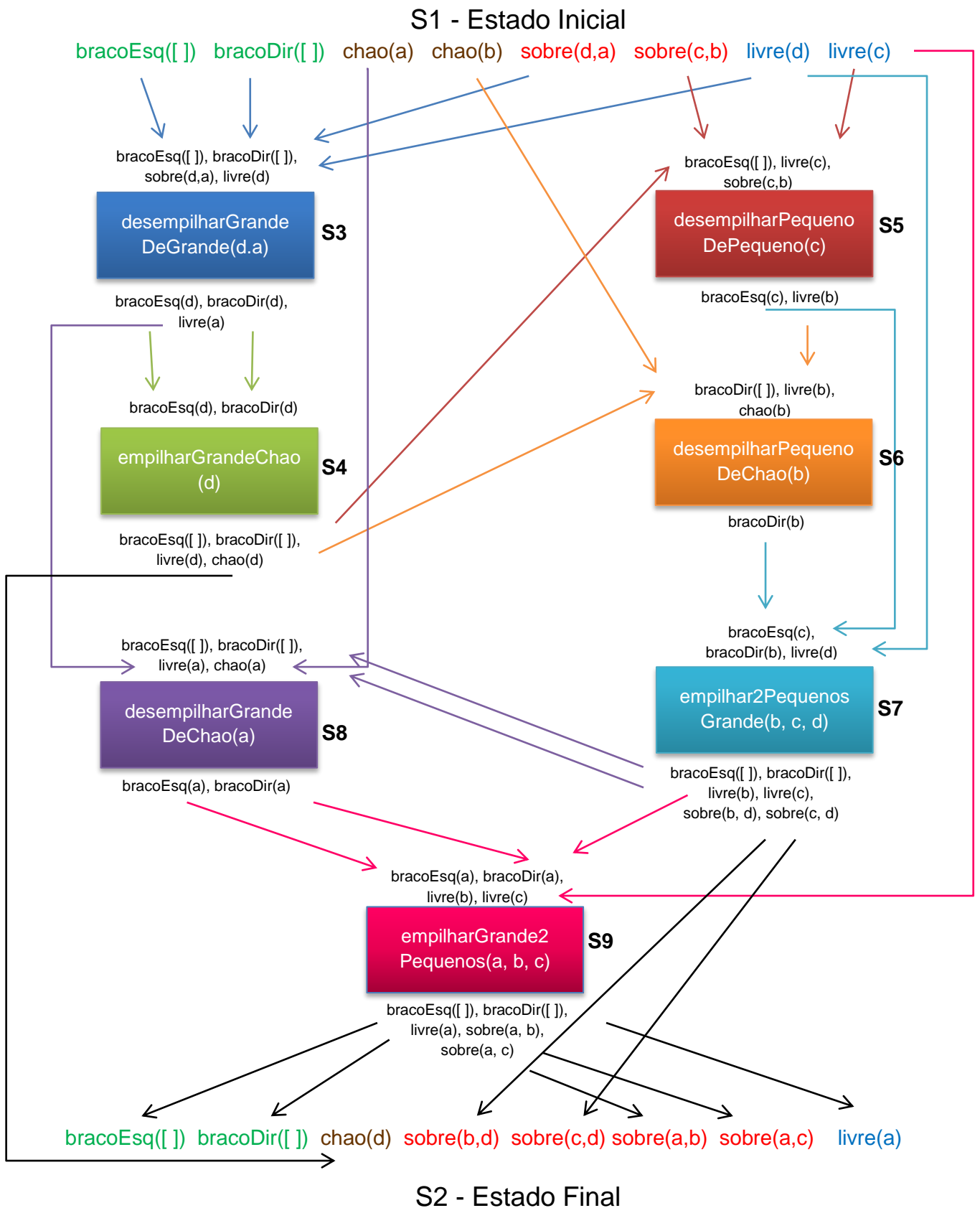
- `acciao(empilharGrande2Pequenos(A, B, C),
[bracoEsq(A), bracoDir(A), livre(B), livre(C)],
[bracoEsq([]), bracoDir([]), livre(A), sobre(A, B), sobre(A, C)],
[bracoEsq(A), bracoDir(A), livre(B), livre(C)])
:- member(A, [a, d]), member(B, [b, c]), A\=B, member(C, [b, c]), C\=B, C\=A.`

Esta ação coloca 1 peça grande da mão do robô para cima de 2 peças pequenas.

2. 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),sobre(b,d),
sobre(c,d), sobre(a,b), sobre(a,c),
livre(a)])`.

3. Pop - Planeador de Ordem Parcial



4. Sequência de ações:

Esta sequência de ações é uma sequência a partir do estado inicial definido até chegar ao estado final. A complexidade é elevada e não consegue assim correr, mas a ordem para o exercício teria de ser:

P = [s1-inicial,
s3- desempilharGrandeDeGrande(d,a),
s4- empilharGrandeChao(d),
s5- desempilharPequenoDePequeno(c),
s6- desempilharPequenoDeChao(b),
s7- empilhar2PequenosGrande(b, c, d),
s8- desempilharGrandeDeChao(a),
s9- empilharGrande2Pequenos(a, b, c),
s2-final]

Conclusão

Após a realização deste trabalho, ficámos a conhecer melhor como funcionam algoritmos de planeamento. 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 este foi resolvido em pouco tempo (apesar de demorar um pouco mais que o exemplo das aulas dado pela professora).

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.