

UNIVERSIDADE DE ÉVORA

Inteligência Artificial
2015/2016

Planeamento



Docente:

Irene Pimenta Rodrigues

Discentes:

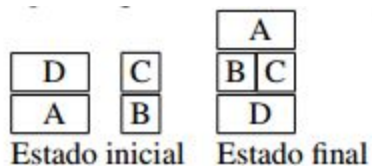
Pedro Jacob - 27421

André Figueira - 31626

Introdução

Neste trabalho foi-nos apresentando um problema que tem a mesma natureza que um problema famoso denominado de “torres de hanoi”. Contudo, as especificações mudaram. Em vez de termos n discos diferentes para n peças, temos apenas 2 dois tipos de peças(blocos pequenos e blocos grandes).

O estado inicial e o estado final é representado no enunciado é o seguinte:



Para tal, vamos mostrar uma das soluções a que chegamos, usando notação Strips.

1. Notação Strips

1.1. Vocabulário

Os diferentes blocos estão identificados da seguinte maneira:

```
bloco_grande (a) .  
bloco_grande (d) .  
bloco_pequeno (c) .  
bloco_pequeno (b) .
```

As acções criadas para o problema em questão foram as seguintes:

```
% tira pequeno de cima de pequeno usando braco direito
acciao(tirarPequeno(D), [dir([]), livre(D), em_cima(D, D1)],
    [dir(D), livre(D1)], [dir([]), em_cima(D, D1), livre(D)]):-
    bloco_pequeno(D), bloco_pequeno(D1), member(D, [c, b]),
    member(D1, [c, b]), D1 \= D.
```

```
% tira pequeno de cima de pequeno usando braco esquerdo
acciao(tirarPequeno(D), [esq([]), livre(D), em_cima(D, D1)],
    [esq(D), livre(D1)], [esq([]), em_cima(D, D1), livre(D)]):-
    bloco_pequeno(D), bloco_pequeno(D1), member(D, [c, b]),
    member(D1, [c, b]), D1 \= D.
```

Esta par de acções pegam num Bloco D que se encontre em cima de outro bloco, obrigando a que este pertença aos grupos dos blocos pequenos. Para pegar neste bloco, é preciso obedecer a um conjunto de regras:

- O braço direito ou esquerdo tem de estar livre.
- D tem de estar livre.
- D tem de estar em cima de D1 (outro bloco pequeno)

No final desta acção, o seguinte tem de acontecer:

- D tem de estar na mão direita, e se não puder, na esquerda.
- D1 tem de estar livre.
- D deixa de estar em cima de D1.
- D deixa de estar livre.
- Braço esquerdo ou direito deixa de estar livre.

```

% tirar bloco do chao usando braco direito
acciao(tirarPequeno(D),[dir([]),livre(D),no_chao(D)],
      [dir(D)],[dir([]),no_chao(D),livre(D)]):-
      bloco_pequeno(D),member(D,[c,b]).

% tirar bloco do chao usando braco esquerdo
acciao(tirarPequeno(D),[esq([]),livre(D),no_chao(D)],
      [esq(D)],[esq([]),no_chao(D),livre(D)]):-
      bloco_pequeno(D),member(D,[c,b]).

```

Esta par de acções pegam num Bloco D que se encontro n chão, obrigando a que este pertença aos grupos dos blocos pequenos. Para pegar neste bloco, é preciso obedecer a um conjunto de regras:

- O braço direito ou esquerdo tem de estar livre.
- D tem de estar livre.
- D tem de estar no chão.

No final desta acção, o seguinte tem de acontecer:

- D tem de estar na mão direita, e se não puder, na esquerda.
- D deixa de estar no chão.
- D deixa de estar livre.
- Braço esquerdo ou direito deixa de estar livre.

```
% meter dois em cima de grande
acao(porDoisPequenos(D,D1,D2),[esq(D),dir(D1),livre(D2)],
    [livre(D),livre(D1),em_cima(D,D2),em_cima(D1,D2),
    dir([],esq([])),
    [esq(D),dir(D1),livre(D2)]]:- bloco_pequeno(D),bloco_pequeno(D1), bloco_grande(D2),
    member(D1,[c,b]),member(D,[c,b]), member(D2,[a,d]), D \= D1.
```

Esta acção permite por dois blocos pequenos (D e D1) em cima de um grande (D2), obrigando a que estes pertençam ao grupo dos blocos pequenos. Para meter este bloco, é preciso obedecer a um conjunto de regras:

- D tem de estar no braço esquerdo
- D1 tem de estar no braço direito.
- D2 tem de estar livre.
- D tem de ser diferente de D1.

No final desta acção, o seguinte tem de acontecer:

- D tem de estar livre.
- D1 tem de estar livre.
- D tem de estar em cima de D2.
- D1 tem de estar em cima de D2.
- O braço esquerdo tem de estar livre.
- O braço direito tem de estar livre.
- D tem de deixar de estar no braço esquerdo.
- D1 tem de deixar de estar no braço direito.
- D2 tem de deixar de estar livre.

```
% usar dois bracos para tirar um grande em cima de outro
acao(tirarGrande(D), [esq([], dir([], livre(D), em_cima(D, D1)],
    [esq(D, dir(D), livre(D1)], [esq([], dir([], livre(D), em_cima(D, D1))]:-
    bloco_grande(D), bloco_grande(D1), member(D, [a, d]), member(D1, [a, d]),
    D \= D1.
```

Esta acção permite usar os dois braços para tirar um bloco grande D de cima de outro bloco grande , obrigando a que este pertença ao grupo dos blocos grandes. Para meter este bloco, é preciso obedecer a um conjunto de regras:

- O braço esquerdo tem de estar livre.
- O braço direito tem de estar livre.
- D tem de estar livre.
- D tem de estar em cima de outro bloco grande (D1).
- D tem de ser diferente de D1.

No final desta acção, o seguinte tem de acontecer:

- D tem de estar no braço esquerdo.
- D tem de estar no braço direito.
- D1 tem de estar livre.
- O braço esquerdo tem de deixar de estar livre.
- O braço direito tem de deixar de estar livre.
- D tem de deixar de estar em cima de D1.

```
% por um bloco grande no chao
acciao(porGrandeChao(D),[esq(D),dir(D)],
      [esq([],dir([]),no_chao(D),livre(D)],[esq(D),dir(D)]):- bloco_grande(D),member(D,[d,a]).
```

Esta acção permite, usando os dois braços, consiga colocar um bloco grande no chão. Para tal tem de obedecer às seguintes regras:

- O bloco tem de estar a ser segurado pelos dois braços.

No final da execução o seguinte tem de acontecer:

- Os braços (esquerdo e direito) estão livres (o bloco deixa de ocupar os dois braços).
- O bloco encontra-se no chão.
- O bloco está livre.


```
% tirar um bloco grande no chao
acao(tirarGrandeChao(D),[esq([],dir([],livre(D),no_chao(D)),
    [esq(D),dir(D)],[livre(D),no_chao(D),esq([],dir([],):- bloco_grande(D), member(D,[d,a])).
```

Esta acção permite que retirar um bloco grande do chão. Como sendo um bloco grande então os dois braços serão usados. Para tal as seguintes regras tem de ser cumpridas:

- Braço esquerdo e direito tem de estar livres.
- O bloco tem de estar livre.
- O bloco tem de estar no chão.

No final o seguinte tem de se realizar:

- O braço esquerdo e direito tem de estar ocupados pelo bloco. Pois sendo um bloco grande necessita dos dois braços, sendo que deixam de estar livres.
- O bloco D deixa de estar livre.
- O bloco D deixa de estar no chão.

```
accao(porGrandeEm2pequenos(D,D1,D2),[livre(D1),livre(D2),esq(D),dir(D)],  
    [esq([],dir([],em_cima(D,D1),em_cima(D,D2),livre(D)],  
    [livre(D1),livre(D2),esq(D),dir(D)]):- bloco_pequeno(D1),bloco_pequeno(D2), bloco_grande(D),  
    member(D2,[c,b]),member(D1,[c,b]), member(D,[a,d]), D1 \= D2.
```

Esta acção permite colocar um bloco grande em cima de dois pequenos (diferentes um do outro). Para tal as seguintes regras tem de ser cumpridas:

- Ambos os braços tem de estar ocupados pelo bloco grande.
- Os blocos pequenos tem de estar livres. Pois neste caso contém-se dois blocos pequenos que estão lado a lado.

No final desta acção o seguinte realiza-se:

- Ambos os braços passam a estar livres.
- O bloco grande passa a estar em cima dos pequenos, logo este passa a estar livre e ambos os pequenos deixam de estar livres.

2. Estado Inicial e Estado Final.

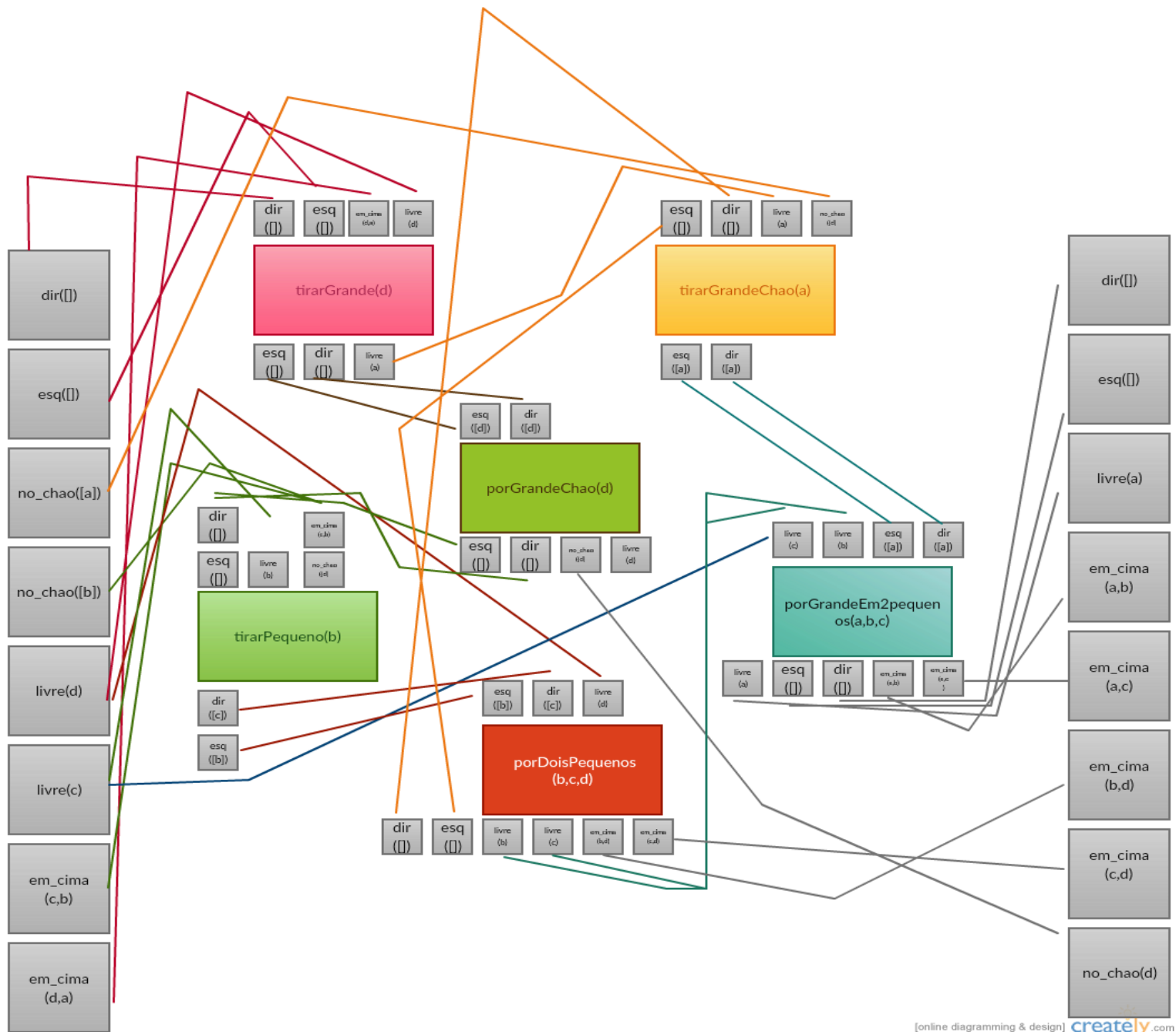
O estado Inicial é representado da seguinte forma.

```
estado_inicial([dir([],esq([],no_chao(a),no_chao(b),livre(d),livre(c),em_cima(c,b),em_cima(d,a))]).
```

O estado final é representado da seguinte forma.

```
estado_final([dir([],esq([],livre(a),em_cima(a,b),em_cima(a,c),em_cima(b,d),em_cima(c,d),no_chao(d))]).
```

3-Planeador de ordem parcial



4. Sequência

Para o problema em questão, a sequência de acções tomada é a seguinte:

1. tirarGrande(d).
2. porGrandeChao(d).
3. tirarPequeno(c).
4. tirarPequeno(b).
5. porDoisPequenos(c,b,d).
6. tirarGrandeChao(a).
7. porGrandeEm2pequenos(a,c,b).

Esta sequência não foi possível se realizar pois o número de passos até encontrar uma sequência demorava bastante tempo e no fim mostrava “Error: Atom table full”. Sabemos que esta sequência não é possível porque, foram testadas todas as operações.

A sequência de 1 até 6, foi executada com sucesso, mas no momento que adicionávamos mais um passo, recebíamos o erro descrito em cima.

Novamente sabemos que o passo 7 funciona pois testado com um caso simples de dois pequenos blocos no chão e por um grande por cima, esse passo sucedeu.

Portanto em baixo está a sequência omitindo o passo 7, mas mantendo a acção no ficheiro:

```
P = [s1-inicial,s9655-tirarGrande(d),s9391-porGrandeChao(d),s9019-tirarPequeno(c),s9656-tirarPequeno(b),s7375-porDoisPequenos(c,b,d),s2-final]
```

5. Conclusão

Após a conclusão do problema que nos foi exposto, podemos afirmar com certeza que entendemos o funcionamento de STRIPS (Stanford Research Institute Problem Solver) além como a necessidade de compreendermos a estrutura de um problema de planeamento e a melhor maneira de resolvermos o mesmo.

Acabamos afirmando sem dúvida que é uma parte indispensável na área de inteligência artificial,, que esperamos explorar mais aprofundadamente no futuro.