UNIVERSIDADE DE ÉVORA

Inteligência Artificial
2015/2016

Planeamento com cálculo de situações

Docente:

Irene Pimenta Rodrigues

Discentes:

Pedro Jacob - 27421

André Figueira - 31626

Introdução

Para começar dando um bocado contexto ao lugar que este trabalho tem na disciplina em questão (Inteligencia Artificial) podemos começar por falar por alto do trabalho anterior, que nos obrigou a explorar a area de Planeamento.

O planejamento é um processo de deliberação que escolhe e organiza ações, antecipando os resultados esperados. Esta deliberação busca alcançar, da melhor forma possível, alguns objetivos pré-definidos.

Esta abordagem de planeamento adopta uma representação em lógica para descrever estados, ações e para definir e computar facilmente o próximo estado. A linguagem básica para esta mesma representação denomina-se de STRIPS (Stanford Research Institute Planning System)

Este ultimo trabalho, pretende explorar mais um bocado este conceito e introduzir um apendice a esta area de conhecimento crucial para a disciplina e para o curso em si. Calculo de situações.

Calculo de situações é descrito como a lógica por detrás das razões das mudanças no mundo, sendo o mesmo descrito por sequencias de situações do estado actual, e por mudanças de uma situação para outra que são causadas por acções

Procedemos então, á apresentaçao da solução do problema apresentado.

1- Notação STRIPS

Acções:

- tirarPequeno(D) : tirar Pequeno de um pequeno usando o braço esq
- tirarPequeno(D) : tirar pequeno do chao usando o braço esq
- tirarGrande(D) : tirar grande de cima de grande
- porGrandeChao(D) : por grande no chao
- tirarGrandeChao(D) : tirar grande do chao
- porPequenosChao(D,D1) : por pequenos no chao
- porGrandeEmGrande(D,D1) : por grane em cima de grande
- tirarPequeno(D) : tirar pequeno do chao usando o braço dir
- tirarPequeno(D) : tirar pequeno de cima de pequeno usando o braço dir

Condições:

- esq(D) : usando o braço esquerdo
- dir(D) : usando o braço direito
- livre(D) : indica que o bloco está livre e pode ser removido usando os braços
- em_cima(D,D1) : bloco D em cima de bloco D1
- no_chao(D) : bloco D está no chão
- bloco_grande(D) : bloco D é um bloco grande
- bloco_pequeno(D) : bloco D é um bloco pequeno

Nota:

- Um bloco só pode ser removido usando os braços se este estiver livre.

- Um bloco grande só pode ser removido se este estiver livre e os dois braços estiverem livres.

- Um bloco D só pode ser colocado por cima de outro D1 se o outro bloco D1 estiver livre, deixando D1 de estar livre e D passa a estar por cima de D1 em que D passa a estar livre.

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

estado_inicial([dir([]),esq([]),livre(d), em_cima(d,a),em_cima(c,b),no_chao(b),livre(c),no_chao(a)]). % estado final sem o ultimo passo
estado_final([dir([]),esq([]),no_chao(d),no_chao(c),em_cima(a,d),em_cima(b,c), livre(b),livre(a)]). % estado final original

% tira pequeno de cima de pequeno usando braco direito
accao(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.

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

% usar dois bracos para tirar um grande em cima de outro
accao(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.

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

% tirar um bloco grande no chao
accao(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]).

% metes os dois pequenos no chao
accao(porPequenosChao(D,D1),
    [esq(D),dir(D1)],
    [esq([]),dir([]),livre(D1),em_cima(D1,D),no_chao(D)],[esq(D),dir(D1)]):-
    bloco_pequeno(D1), D1 = b, bloco_pequeno(D), D = c.

accao(porGrandeEmGrande(D,D1),[esq(D),dir(D),livre(D1)],[em_cima(D,D1),esq([]),dir([]),livre(D)],[livre(D1),esq(D),dir(D)]):-
bloco_grande(D),D = a, bloco_grande(D1) , D1 = d.

% tira pequeno de cima de pequeno usando braco esquerdo
accao(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.

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

2. Represente o estado inicial deste problema com o vocabulário definido na alínea anterior no calculo de situações.

```
% estado inicial

h(dir([]),S0).
h(esq([]),S0).
h(em_cima(d,a),S0).
h(em_cima(c,b),S0).
h(livre(c),S0).
h(livre(d),S0).
h(no_chao(a),S0).
h(no_chao(b),S0).
```

3. Descreva as consequências positivas de cada acção no cálculo de situações.

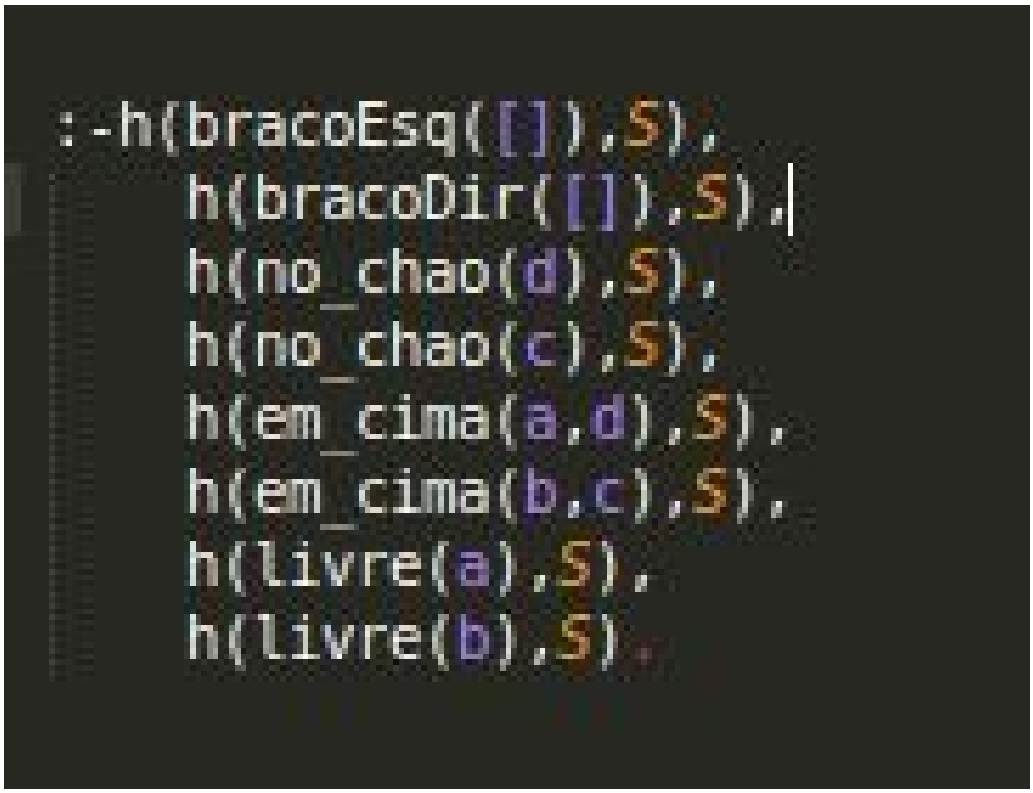
```
blocos2.pl
% consequencias
pgc(D):- no_chao(D),livre(D),esq([]),dir([]).
ppc(D,D1) :- esq([]),dir([]),no_chao(D1),livre(D),
    em_cima(D,D1),bloco_pequeno(D),bloco_pequeno(D1),
    D \= D1.

pgg(D,D1):- esq([]), dir([]), em_cima(D,D1), livre(D).

tgc(D):- esq(D),dir(D).

tgg(D):- esq(D),dir(D),livre(D1).


% tirar pequeno do chao braço dir
h(dir(D),r(tirarPequeno(D),S)):-
    h(dir([]),S),
    h(no_chao(D),S),
    h(livre(D),S),
    bloco_pequeno(D).


% tirar pequeno do chao usando braço esq
h(esq(D),r(tirarPequeno(D),S)):-
    h(esq([]),S),
    h(no_chao(D),S),
    h(livre(D),S),
    bloco_pequeno(D).

% tirar pequeno de outro pequeno usando braço dir
h(dir(D),r(tirarPequeno(D),S)):-
    h(dir([]),S),
    h(em_cima(D,D1),S),
    h(livre(D),S),
    bloco_pequeno(D),bloco_pequeno(D1),D \= D1.

% tirar pequeno de outro pequeno usando braço esq
h(esq(D),r(tirarPequeno(D),S)):-
    h(esq([]),S),
    h(em_cima(D,D1),S),
    h(livre(D),S),
    bloco_pequeno(D),bloco_pequeno(D1), D1 \= D.


% tirar grande do chao usando os dois braços
h(tgc(D),r(tirarGrandeChao(D),S)):-
    h(esq([]),S),
    h(dir([]),S),
    h(no_chao(D),S),
    h(livre(D),S),
```

```
% tirar grande cima de outro grande usando os dois braços
h(tgg(D),r(tirarGrande(D),S)):-
    h(esq([]),S),
    h(dir([]),S),
    h(em_cima(D,D1),S),
    h(livre(D),S),
    bloco_grande(D),
    bloco_grande(D1),
    D \= D1.


% por grande no chao usando os dois braços
h(pgc(D),r(porGrandeChao(D),S)):-
    h(esq(D),S),
    h(dir(D),S),
    bloco_grande(D).


% por um pequeno no chao e outro em cima do outro pequeno
% usando os blocos que estao nos braços
h(ppc(D,D1),r(porPequenosChao(D,D1),S)):-
    h(esq(D),S),
    h(dir(D1),S)
    bloco_pequeno(D),
    bloco_pequeno(D1),
    D \= D1.

% por grande em cima de outro grande
h(pgg(D,D1),r(porGrandeEmGrande(D,D1),S)):-
    h(esq(D),S),
    h(dir(D),S),
    h(livre(D1),S),
    bloco_grande(D),
    bloco_grande(D1),
    D \= D1.
```

4. Descreva as regras de inercia para cada fluente (condição) no cálculo de situacões.

```
% inercia

h(esq(D),r(D1,S)):-
    h(esq(D),S),
    D1 \= tirarPequeno(D).

h(dir(D),r(D1,S)):-
    h(dir(D),S),
    D1 \= tirarPequeno(D).

h(dir(D),esq(D),r(D1,S)):-
    h(esq(D),dir(D),S),
    D1 \= tirarGrandeChao(D),
    D1 \= tirarGrande(D).


h(esq([]), dir([]), r(D1, S)) :-
    h(esq([]), S),
    h(dir([]), S),
    X \= porGrandeEmGrande(D,A),
    X \= porGrandeChao(D).

h(esq(D),dir(D1),r(D2,S)):-
    h(esq(D),S),
    h(dir(D1),S),
    X \= porPequenosChao(D,D1).
```

5. Indique a query que usou para obter o plano para este problema



6. indique o plano que obteve com as suas definições.

```
S =
    r(porPequenosChao(c,b),
    r(tirarPequeno(c),
    r(tirarPequeno(c),
    r(porGrandeEmGrande(a,d),
    r(tirarGrandeChao(a),
    r(porGrandeChao(d),
    r(tirarGrande(d),S0))))))).
```

Conclusão

Durante a realização deste trabalho, viemos-nos a aperceber que Calculo de situações completa conhecimentos anteriores que havíamos obtidos nesta disciplina e permite-nos assim resolver problemas de maneira mais completa, robusta e correta.

Continuamos a concluir, que após a realização deste trabalho ficamos com uma visão muito mais ampla da área da logica do calculo de situações. Mais importante ainda, devido ao teor prático elevado da mesma, iremos no futuro ser capazes de usar estes mesmo conhecimentos em termos práticos para possíveis projectos futuros.

Fazendo um apanhado final da disciplina (visto este ser o ultimo trabalho de um grupo de eles) podemos também dizer com certeza que nos preparou bem para possíveis avanços em termos de conhecimento da nossa parte na área de inteligencia artificial, e , mais importante, que nos mostrou um lado da área em questão que não tínhamos sequer a noção de existir.