

Inteligência Artificial

Trabalho 5

Planeamento



Discente:

André Gouveia nº 26918

Évora 2015

Índice

Introdução.....	2
1- Notação Strips- Vocabulário.....	2
2- Estado inicial e estado final.....	3
3- Resolução do pop para transição de estados 0-1	3
4- Sequência de acções obtidas do pop	4
5- Execução.....	4

Introdução

Para este trabalho é pedido o planeamento de operações de um robot com 2 braços colocado em frente a uma fila de 4 copos. O mesmo pode pegar em 2 copos adjacentes e roda-los.

O objectivo do planeamento é, dado um estado inicial dos copos e um estado final pretendido para esses mesmo copos, definir quais os passos que o robot deverá seguir.

Para o efeitos de projecto, definimos que o robot começa frente aos copos 1 e 2 e que os copos estão todos virados para cima. O estado final o robot está em frente aos mesmos copos (1 e 2) estando os mesmos virados para baixo e os restantes para cima.

1- Notação Strips- Vocabulário

- Condições

pcima(C) - copo virado para cima.

pbaixo(C) - copo virado para baixo.

mao(C1,C2) - copos C1 e C2 na mão.

mvazia - mão não tem copos.

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 baixo 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.

As acções são definidas da seguinte forma:

`accao(NomeAccao,[condições],[addList],[removeList]).`

`accao(desloca(C1,C2),[mvazia, frente(C3,C4)],[frente(C1,C2)],[frente(C3,C4)]).`

`accao(agarra(C1,C2),[mvazia, frente(C1,C2)],[mao(C1,C2)],[mvazia]).`

`accao(viraCC(C1,C2),[mao(C1,C2),pbaixo(C1),pbaixo(C2)],[pcima(C1),pcima(C2),mvazia],[mao(C1,C2),pbaixo(C1),pbaixo(C2)]).`

`accao(viraBB(C1,C2),[mao(C1,C2),pcima(C1),pcima(C2)],[pbaixo(C1),pbaixo(C2),mvazia],[mao(C1,C2),pcima(C1),pcima(C2)]).`

`accao(viraCB(C1,C2),[mao(C1,C2),pbaixo(C1),pcima(C2)],[pcima(C1),pbaixo(C2),mvazia],[mao(C1,C2),pbaixo(C1),pcima(C2)]).`

`accao(viraBC(C1,C2),[mao(C1,C2),pcima(C1),pbaixo(C2)],[pbaixo(C1),pcima(C2),mvazia],[mao(C1,C2),pcima(C1),pbaixo(C2)]).`

2- Estado inicial e estado final

`estado_inicial([pbaixo(1),pbaixo(2),pbaixo(3),pbaixo(4),mvazia,adjacente(1,2),adjacente(2,3),adjacente(3,4), frente(1,2)]).`

`estado_final([pcima(1),pcima(2),pcima(3),pbaixo(4),mvazia,adjacente(1,2),adjacente(2,3),adjacente(3,4), frente(1,2)]).`

3- Resolução do pop para transição de estados 0-1

Para a primeira resolução o pop define o estado inicial como s1 e o final como s2.

Procura no estado final uma condição que não esteja ainda feita e a acção que possibilita obter essa condição. Essa acção gera um novo estado e o pop irá verificar se as condições para esse estado estão definidas e assim sucessivamente até ter os passos todos até à condição final estar satisfeita. Caso exista uma situação de conflito, o link/acção fica marcado como tendo de ocorrer antes da acção que causa o conflito. Ex:

$X = [s1\text{-inicial}, s35\text{-desloca}(1,2), s4\text{-agarra}(1,2), s3\text{-viraCC}(1,2), s2\text{-final}]$

É uma solução do pop e o s35 é uma acção que tem de ser feita antes da s3.

4- Sequência de acções obtidas do pop

$X = [s1\text{-inicial}, s4\text{-agarra}(1,2), s3\text{-viraCC}(1,2), s2\text{-final}]$

5- Execução

Para correr o programa basta estar na pasta do ficheiro pop.pl e consultar o mesmo em gnu prolog. Depois de consultado, fazer a pesquisa : plano(X).

X será a sequência de acções a fazer para o resultado pretendido.