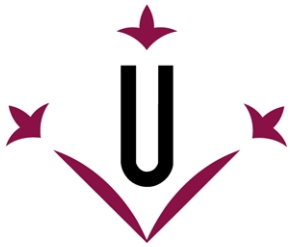


Universitat de Lleida

Enginyeria Informàtica

Aprenentatge i Raonament Automàtic

Homework Assignment 1: Knowledge Based Agents

Alba Lamas Varela
Humbert Vallés Teixidó

Professorat:
Ramón Béjar Torres

24 d'abril de 2017

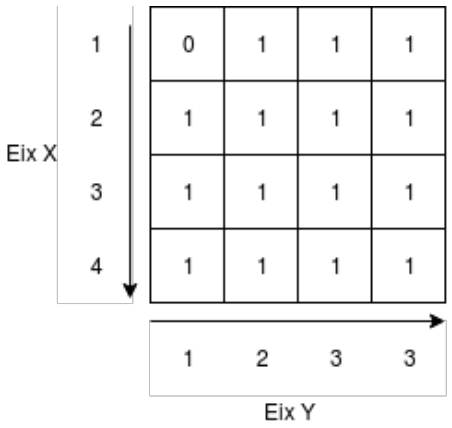
Índex

1 Introducció

L'objectiu d'aquesta pràctica és la implementació del *BarcenasWorld problem* tractat a classe. Per fer-ho s'ha creat un programa en Python que donada una determinada dimensió del món, i un seguit de passos a donar, crea un programa en prolog el qual mostra "el món de Barcenas"amb les possibles posicions en les que es pot trobar tenint en compte el coneixement del món obtingut a partir del tractament dels passos.

2 BarcenasWorld

El *BarcenasWorld* sempre tindrà dimensió NxN, sent l'eix X les files i l'eix Y les columnes, i numerant les posicions començant des de dalt a l'esquerra. A continuació es mostra una representació gràfica de l'estat inicial del BarcenasWorld en una modelització de mida 4x4.



L'estat inicial ens dóna una representació del *BarcenasWorld* abans d'accedir-hi, i per tant abans de tenir qualsevols tipus d'informació. Així doncs, tindrà totes les posicions a 1, execepte la (1,1), la qual es trobarà a 0.

3 WorldGenerator.py

Programa en python que donada una entrada determinada crea un programa en prolog amb les clàusules necessàries per resoldre l'instància del problema amb la que estem tractant donades les dades.

3.1 Input

L'entrada consistirà en:

1. Un número positiu, n, el qual serà la dimensió (nxn) del *Barcenas World* en el qual conviu el nostre agent amb Bárcenas, Mariano y Cospedal.

2. Una llista, que representa una seqüència de l passos, els quals segueixen l'estructura següent:

   $$[[x_1, y_1, s_1, ma_1, ca_1], [x_2, y_2, s_2, ma_2, ca_2], ..., [x_l, y_l, s_l, ma_l, ca_l]]$$

   Sigui X i Y les coordenades relatives a la posició en la que ens trobem actualment, s la lectura del sensor d'olor, i ma i ca l'informació relativa a Mariano i Cospedal.

   Els passos han d'estar en el format que es mostra a continuació: una llista de llistes sense espais.

   $$[[3, 3, 0, 1, 1], [3, 4, 0, -1, -1]]$$

   També es pot passar com el nom d'un fitxer .txt on estiguin els passos guardats amb el següent format:

   3 3 0 1 1
   3 4 0 -1 -1

   Separats per espais, sense comes i entre pas i pas un salt de línia.

   Tal i com indica a l'enunciat de la pràctica, x_n i y_n són la posició X i Y on està l'agent; s_n l'estat del sensor d'olor que té l'agent en aquella posició, el qual indica 1 si olora a Bárcenas i 0 si no; ma_n indica en cas d'estar Mariano en aquella coordenada, si Bárcenas es troba a la seva esquerra (1), o a la seva dreta (0) i en cas de que Mariano no es trobi aquella posició, ho indicarem amb un -1. Per últim, co_n, indica si ens trobem a Cospedal en la posició actual. En cas afirmatiu ens dirà si Mariano menteix (1) o si diu la veritat (0). En cas de que no estigui en aquell coordenada, s'indicarà amb un -1.

3.2 Output

Un exemple de sortida del programa, tenint com a entrada:

1. n = 4

2. steps = [[3,3,0,1,1],[3,4,0,-1,-1]]

seria el següent:

```
Estado final:
[0,0,1,1]
[0,0,0,0]
[0,0,0,0]
[0,0,0,0]
```

Aprofitem aquest apartat per comentar que en alguns ordinadors ens apareix el següent warning al accedir al entorn de prolog:

```
Warning: /home/alba/AIRA/BarcenasWorld/init.pl:1:
Using a non-error value for unknown in the global module
causes most of the development environment to stop working.
Please use :- dynamic or limit usage of unknown to a module.
See http://www.swi-prolog.org/howto/database.html
```

3.3 Execució BarcenasWorld.pl

Un cop escrit tot el programa prolog, aquest serà executat des del mateix programa python mitjançant la comanda següent:

```
'swipl -q -f init.pl -s BarcenasWorld.pl -g
"updateSequenceOfSteps(' SITUACIOINICIAL, STEPS, FS),halt"'
```

on SITUACIOINCIAL és una llista de llistes que modelitza l'estat inicial, tal com ha estat comentat en el segon apartat i STEPS està format pel conjunt de passos que s'han passat a l'entrada del programa de python.

4 Relacions de les normes en CP1

Per tal de fer l'implementació del *BarcenasWorld* en prolog ha estat necessari crear la funció recursiva updateSequenceOfSteps, la qual rep un seguit de passos a realitzar, un estat inicial i guarda el resultat d'executar els passos en un estat final. La funció agafa el primer element de la llista de passos i obté els elements que necessita, en aquest cas les posicions X i Y, el resultat del sensor d'olor i què diu Mariano. Aquestes són les dades que enviem a la funció updatePosBarcenasLocs, juntament amb què diu Cospedal i on volem guardar el resultat d'aplicar el pas fet. Tot seguit es fa una crida a la mateixa funció, sent els passos els que queden a la llista. Seguirem fent crides fins obtenir una llista buida. Llavors, imprimirem l'estat final i acabarem.

```
updateSequenceOfSteps( FS, [], FS ):- write( 'Estado final:
' ), writeFinalState(FS), nl.

updateSequenceOfSteps( PrevLocs, [H|T], FS )
    :-
        set( 1, Lies ),
        nth0(0, H, X),
        nth0(1, H, Y),
        nth0(2, H, S),
        nth0(3, H, M),
        updatePosBarcenasLocs( PrevLocs, X, Y, S, M, Lies,NextLocs ),
        updateSequenceOfSteps( NextLocs, T, FS ).
```

La funció updatePosBarcenasLocs ha estat modificada per tal de que a més de fer intersectLocs amb la posició de l'agent i la lectura del sensor d'olor, també ho faci amb el resultat que obtenim del primer intersectLocs i el resultat que obtenim a partir del que diu Mariano sobre la posició de Barcenas.

ORIGINAL:

```
updatePosBarcenasLocs( PrevLocs, AgentPosX, AgentPosY,  SmellXY, FinalLocs )
   :-
      isBarcenasAround( AgentPosX, AgentPosY, SmellXY, NewLocs ),
      intersectLocs( PrevLocs, NewLocs, FinalLocs ), !,
      write( 'Estado resultante: ' ), write( FinalLocs ), nl.
```

MODIFICADA:

```
updatePosBarcenasLocs( PrevLocs, AgentPosX, AgentPosY,  SmellXY,
MarianoXY, Cospedal, FinalLocs )
   :-
      isBarcenasAround( AgentPosX, AgentPosY, SmellXY, AfterSmell ),
      intersectLocs( PrevLocs, AfterSmell, Locs ), !,
      isBarcenasOnLeft( AgentPosX, AgentPosY, MarianoXY, MarianoLocs ),
      intersectMarianoLies( MarianoXY, Cospedal, MarianoLocs, NewLocs ),
      intersectLocs( Locs, NewLocs, FinalLocs ), !,
      write( 'Estado resultante: ' ), writeFinalState(FinalLocs), nl.
```

Les clàusules que utilitzem per tal de fer l'intersectRow i l'intersectLocs a partir del diu Mariano segueixen el mateix esquema que les del intersectLocs del

programa original, simplement hem susbtituït els noms i algunes dades per les que ens interessen. La funció que canvia més és la d'intersectLies, ja que depèn del que diuen Mariano i Cospedal

```
intersectLies( -1, X, 1, 1).
intersectLies( M, 1, 1, 0).
intersectLies( M, 1, 0, 1).
intersectLies( M, 0, X, X).
intersectLies( M, -1, Y, Y).

intersectRowMarianoLies( M, Lies, [], [] ).
intersectRowMarianoLies( M, Lies, [PH|PT], [FH|FT] ) :-
            intersectLies( M, Lies, PH, FH ),
            intersectRowMarianoLies( M, Lies, PT, FT ).

intersectMarianoLies( M, Lies, [], [] ).
intersectMarianoLies( M, Lies, [PrevRow|PrevLocs], FinalLocs ) :-
            intersectRowMarianoLies( M, Lies, PrevRow, FinalRow ),
            intersectMarianoLies( M, Lies, PrevLocs, RestOfRows ),
            FinalLocs = [ FinalRow | RestOfRows ].
```

Funciona de forma similar al intersectLocsInfo: per cada posició, es calcula un valor resultant a partir de la intersecció de les respostes de Cospedal i Mariano.

Primer comprovem si ens hem trobat a Mariano en aquell pas, i si ha dit alguna cosa. En cas que no ens l'haguem trobat o no contesti res ens retorna -1. En cas de rebre -1 per part de Mariano, considerem que totes les posicions són 1, per això, digui el que digui la Copedal (X a la funció intersectLies), rebrem un 1, i retornem 1.

En els altres casos, digui el que digui Mariano, si Cospedal diu que menteix (un 1 a la segona entrada de la funció intersectLies), donarem a l'última variable el valor contrari del que s'havia calculat en aquella posició a partir de la resposta de Mariano.

En cas de que Cospedal digui que Mariano no menteix, es dóna a l'última variable el mateix valor que tenia la variable basada en la resposta de Mariano en aquella posició.

5 Dificultats i solucions

- La principal dificultat amb la que ens vam trobar va ser considerar la resposta de Cospedal. Com que diu si Mariano menteix o no en el pas en la que ens la trobem, no sabiem com guardar la informació de Mariano fins trobar-la i després tractar-a (en prolog). Per solucionar-ho en el programa python es busca si ens hem trobat a Cospedal o no.

  En cas de trobar-la, com la informació que ens dóna és consistent (no pot dir que no menteix en un pas i després que no, o al revès) es guarda una variable de python amb la informació (0, 1 o -1). Des de python s'escriu una línia al programa prolog, a la funció principal com la següent (sigui X 0, 1 o -1):

  ```
  set(X, Lies),
  ```

  Aquesta funció fa que la variable Lies adopti el valor respost per Cospedal. Més tard, s'agafa el mapa resultant de la resposta de Mariano, i es conserven els valors, o es fiquen els contraris, segons la resposta de Cospedal.

  Som conscients que si l'implementació fos correcta, la lectura del lies s'hauria de fer en el programa de prolog i només aplicar els resultats de les respostes obtingudes per Mariano un cop ens trobem a Cospedal, però no hem estat capaços d'implementar-ho a causa del poc coneixement que tenim del llenguatge prolog.

- No es tracta exactament d'una dificultat però creiem que és important mencionar-ho: som conscients de que la modelització que hem fet del *BarcenasWorld* no és la mateixa que s'ha fet a classe pel que fa a la distribució del eixos de coordenades (la qual es troba invertida) i la numeració de les posicions (la qual tal com ha estat comentat en el segon apartat, comença des de dalt a l'esquerra enlloc de des de baix a l'esquerra), però vam fer tota la implementació seguint aquest model i ens en vam adonar un cop finalitzat el codi, així que vam decidir mantenir-ho.

6 Tests de proves

6.1 Exemple d'execució

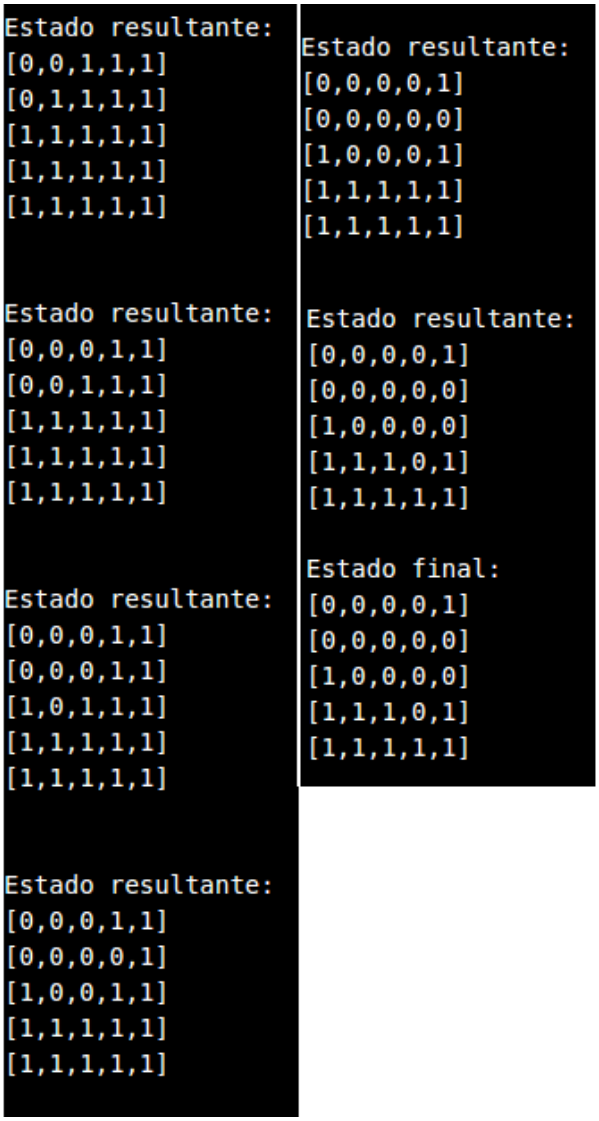
A continuació es mostra un exemple d'execució del programa passant la llista de passos com a paràmetre. Si es trobés en un fitxer simplement canviariem la llista pel nom del fitxer.

```
./WorldGenerator n [[X,Y,S,M,C]...]

./WorldGenerator n foo.txt
```

```
silmar@silmar-Aspire-F5-573G ~/Desktop/udl/aprenentatge/BarcenasWorld $ ./WorldGenerator.py 6 [[3,3,0,1,1],[3,4,0,-1,-1]]
Warning: /home/silmar/Desktop/udl/aprenentatge/BarcenasWorld/init.pl:1:
        Using a non-error value for unknown in the global module
        causes most of the development environment to stop working.
        Please use :- dynamic or limit usage of unknown to a module.
        See http://www.swi-prolog.org/howto/database.html

Estado resultante:
[0,0,1,1,1,1]
[0,0,0,1,1,1]
[0,0,0,0,1,1]
[0,0,0,1,1,1]
[0,0,1,1,1,1]
[0,0,1,1,1,1]


Estado resultante:
[0,0,1,1,1,1]
[0,0,0,0,1,1]
[0,0,0,0,0,1]
[0,0,0,0,1,1]
[0,0,1,1,1,1]
[0,0,1,1,1,1]

Estado final:
[0,0,1,1,1,1]
[0,0,0,0,1,1]
[0,0,0,0,0,1]
[0,0,0,0,1,1]
[0,0,1,1,1,1]
[0,0,1,1,1,1]

silmar@silmar-Aspire-F5-573G ~/Desktop/udl/aprenentatge/BarcenasWorld $
```

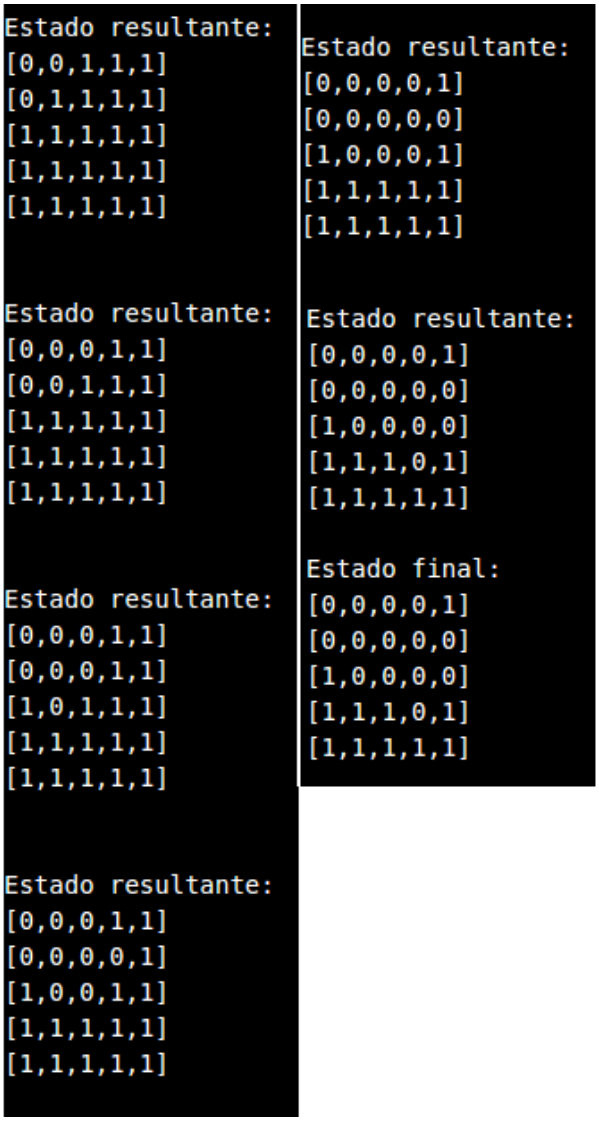
6.2 Prova 1

Món de 5x5 en el qual el **sensor d'olor no detecta** a Bárcenas. Passos donats:



```
1 1 0 -1 -1
1 2 0 -1 -1
2 2 0 -1 -1
2 3 0 -1 -1
2 4 0 -1 -1
3 4 0 -1 -1
```

RESULTAT:

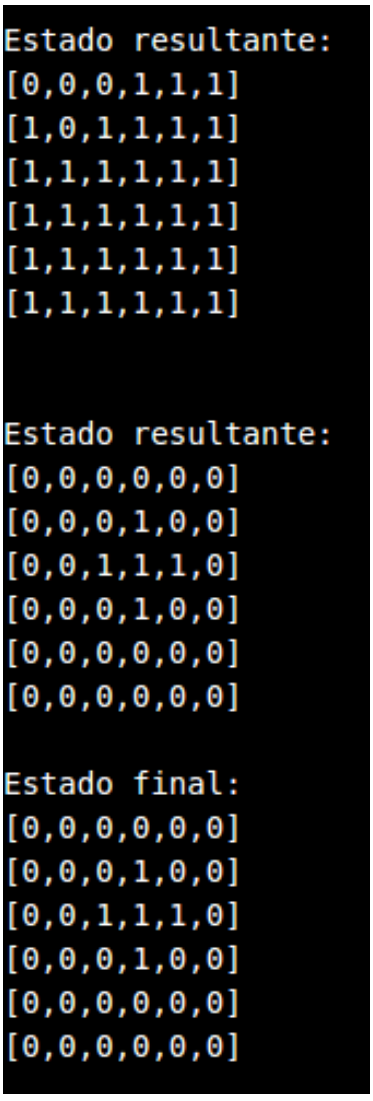


6.3 Prova 2

Món de 6x6 en el qual el **sensor d'olor detecta** a Bárcenas, eliminant totes les altres posicions. Passos donats:

```
1 2 0 -1 -1
3 4 1 -1 -1
```

RESULTAT:

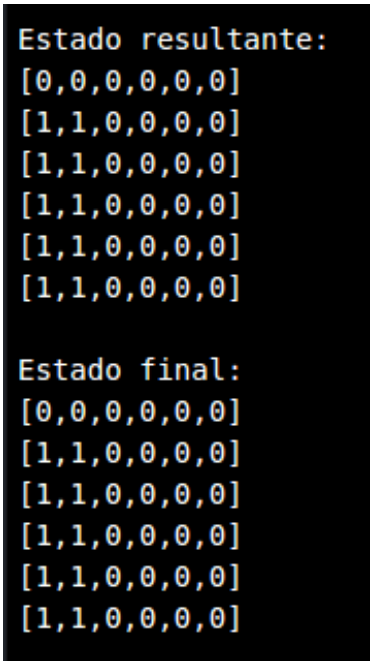


6.4 Prova 3

Món de 6x6 en el qual ens trobem a Mariano i ens indica que Bárcenas està a la seva **esquerra** i **no menteix**. Passos donats:

```
1 3 0 1 0
```

RESULTAT:

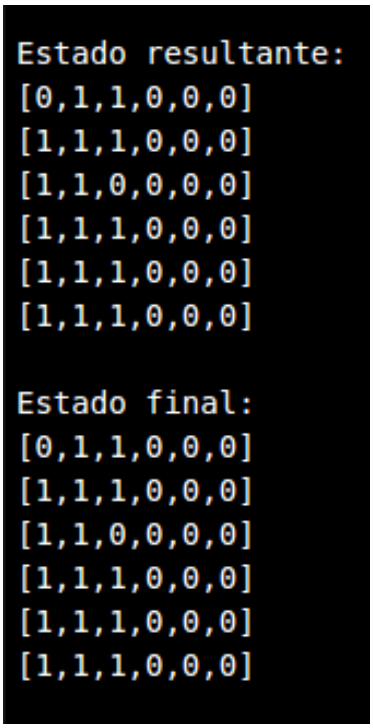


6.5 Prova 4

Món de 6x6 en el qual ens trobem a Mariano i ens indica que Bárcenas està a la seva **dreta** i **no menteix**. Passos donats:

```
3 4 0 0 0
```

RESULTAT:

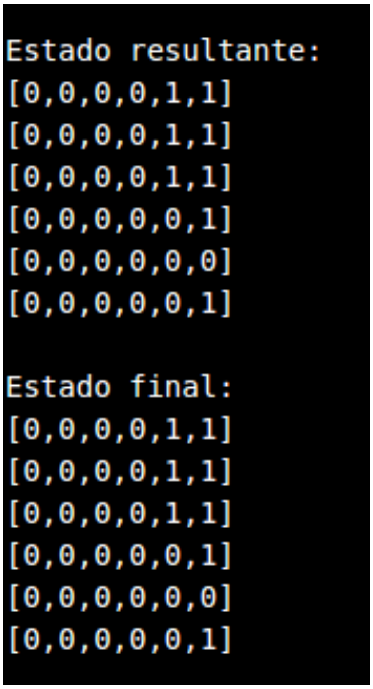


6.6 Prova 5

Món de 6x6 en el qual ens trobem a Mariano i ens indica que Bárcenas està a la seva **esquerra** i **menteix**. Passos donats:

```
5 5 0 1 1
```

RESULTAT:

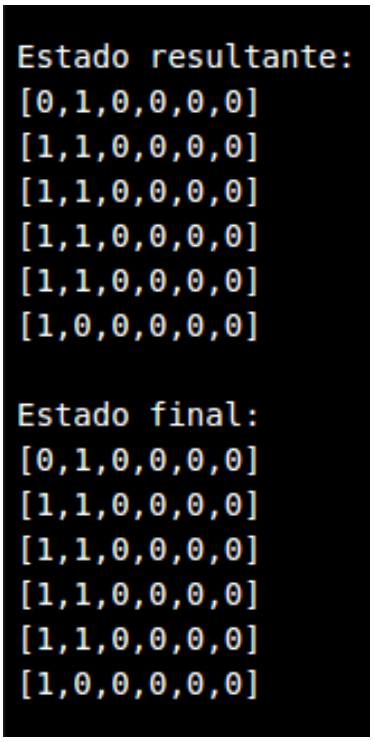


6.7 Prova 6

Món de 6x6 en el qual ens trobem a Mariano i ens indica que Bárcenas està a la seva **dreta** i **menteix**. Passos donats:

```
6 3 0 0 1
```

RESULTAT:

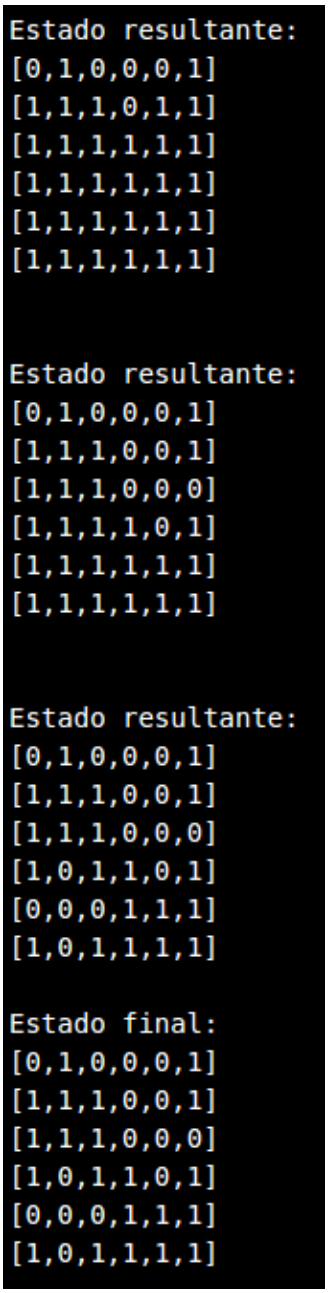


6.8 Prova 7

Món de 6x6 en el qual ens trobem a Cospedal i no a Mariano. Passos donats:

```
1 4 0 -1 1
3 5 0 -1 1
5 2 0 -1 1
```

RESULTAT:



6.9 Prova 8

Món de 6x6 en el qual ens trobem a Mariano més d'un cop. Passos donats:

```
6 2 0 0 0
5 4 0 0 0
3 5 0 0 0
```

RESULTAT:

```
Estado resultante:
[0,1,1,1,1,1]
[0,1,1,1,1,1]
[0,1,1,1,1,1]
[0,1,1,1,1,1]
[0,0,1,1,1,1]
[0,0,0,1,1,1]


Estado resultante:
[0,0,0,1,1,1]
[0,0,0,1,1,1]
[0,0,0,1,1,1]
[0,0,0,0,1,1]
[0,0,0,0,0,1]
[0,0,0,0,1,1]


Estado resultante:
[0,0,0,0,1,1]
[0,0,0,0,0,1]
[0,0,0,0,0,0]
[0,0,0,0,0,1]
[0,0,0,0,0,1]
[0,0,0,0,1,1]

Estado final:
[0,0,0,0,1,1]
[0,0,0,0,0,1]
[0,0,0,0,0,0]
[0,0,0,0,0,1]
[0,0,0,0,0,1]
[0,0,0,0,1,1]
```