Universitat de Lleida

Universitat de Lleida

Grau en Enginyeria Informàtica

Aprenentatge i Raonament Automàtic

Pràctica 1, Agents Intel·ligents en CP1

Autors:
Jordi Ricard Onrubia Palacios
Marcel Porta Valles

Professor:
Ramón Bejar Torres

24 d'abril de 2017

Resum

En el següent document s'explica la metodologia emprada per a la cerca de Bàrcenas en un món de NxN, on l'agent que realitza la cerca rebrà informació per part de tres fonts diferents, la primera un Sensor d'Olor que indicarà si Bàrcenas està en posicions adjacents, Rajoy que podrà donar indicacions sobre on pot estar Bàrcenas i per últim Cospedal que ens indicarà si la informació donada per Rajoy és correcta o no, per tal de realitzar aquesta cerca es fa servir lògica de primer ordre, i el llenguatge a utilitzar serà Prolog (encara que el programa en Prolog es programa amb un programa en Python).

Índex

1 Execució del Programa:

Per a executar el programa en Prolog inicialment s'ha d'executar el programa en Python, ja que serà aquest que, depenent de les dimensions, crearà un programa en Prolog que s'adeqüi.

Execució Python:

./findingBarcenas.py <N, Seqüència de Passos>

On:
N: Indica la mida del món en el qual em de buscar a Bàrcenas, és un nombre natural.
Seqüència de Passos: és una llista formada de llistes que contenen els passos que s'han de seguir amb el següent format.

[X,Y, SmellsXY,MaXY, CaXY]

On:
X: Indica la coordenada de les X del món.
Y: Indica la coordenada de les Y del món.
SmellsXY: Indica la resposta que dóna el detector d'olor, 1 si Bàrcenas està en una de les posicions adjacents 0 si no.
MaXY: Indica la resposta donada per Rajoy sobre els possibles estats de Bàrcenas, 0 per indicar que es troba a l'esquerra, 1 per indicar que es trobi o en la mateixa columna o a la dreta de la seva posició i -1 si encara no l'hem trobat.
CaXY: Indica la resposta donada per Cospedal sobre si Rajoy diu la veritat, 0 per indicar que no menteix, 1 per indicar que menteix i -1 en cas que no l'hàgim trobat.

Un cop finalitzada l'execució d'aquest es realitzarà l'execució del programa en Prolog donant la solució així com cada un dels estats del món després de cada pas, a més a més es generà un fitxer anomenat findingBarcenasNxN.pl, on N és la dimensió d'aquell món, aquest contindrà el codi necessari per executar qualsevol seqüència de passos sempre que les posicions X Y que s'indiquin als passos corresponguin a les dimensions del món, és a dir $1 \leq X, Y \leq N$, és important tenir en compte les dimensions, ja que sinó, no funcionarà.

Execució del programa en Prolog.

```
swipl -q -f findingBarcenasNxN.pl -t 'execofSteps([Món],
[Seqüència de passos],_,_,_,_)'
```

On: Món: Taulell que representa el món en el qual hem de trobar a Bàrcenas amb el següent format.

```
[[x1y1,x1y2,...,x1yN],[x2y1,x2y2,...,x2yN],[x3y1,x3y2,...,x3yN]
,...,[xNy1,xNy2,...,xNyN]]
```

Seqüència de passos: Passos que es duran a terme durant la cerca, amb el format prèviament anomenat.

2 Funcions:

2.1 execSeqofSteps:

Funció recursiva que cridarà updateSequenceOfSteps per tal d'executar tots els passos, aquesta finalitzarà quan la llista de passos es quedi buida.
Arguments:

- PrevLocs: Matriu NxN que representa el món en el qual em de trobar a Bàrcenas, aquesta inicialment serà una matriu tot a uns a excepció de la posició 1,1 que serà 0 sent aquesta la posició inicial.
- Llista de passos: Llista de passos a seguir per tal de buscar a Bàrcenas, aquesta es compon de X, Y direccions a les quals ens anem desplaçant per a realitzar la cerca, S, detector d'olor de Bàrcenas, M i C, respostes donades per Mariano i Cospedal.
- FinalLocs: Matriu NxN actualitzada amb les noves possibles localitzacions de Bàrcenas després d'executar cada un dels passos de la llista.
- Pas[]: Aquests arguments ens permeten guardar la Y on es va trobar Rajoy, la seva resposta així com la resposta de Cospedal.

2.2 updateSequenceOfSteps:

Funció que s'encarrega de processar el pas passat per argument, aquest executarà totes les altres funcions per tal de realitzar aquest procés.
Arguments:

- PrevLocs: Matriu NxN que representa el món en el qual em de trobar a Bàrcenas, aquesta inicialment serà una matriu tot a uns a excepció de la posició 1,1 que serà 0 sent aquesta la posició inicial.
- SequenceOfSteps: Passos a realitzar en aquell moment, aquests es componen de les posicions X, Y del món on hem de buscar a Bàrcenas, S detecto d'olor, M Mariano i C Cospedal.
- Pas[]: Argument on es passen els estats previs per a la columna on hem trobat a Mariano, la resposta de Mariano i la resposta de Cospedal.
- Fut[]: Argument on es guardarà la Columna on hem trobat a Mariano en cas que el trobem, la resposta de Mariano si l'hem trobat, en cas contrari guardarem -1, i la resposta de Cospedal si l'hem trobat, en cas contrari guardem -1.
- FinalLocs: Estat del món un cop realitzat el pas passat per argument amb les possibles localitzacions de Bàrcenas.

2.3 isBarcenasAround:

En aquest apartat s'escriuen totes les possibilitats dels estats de les caselles donat que el detector d'olor dona 1 o 0, la funció s'encarrega d'escollir l'adequat a l'hora de passar els arguments per tal de retornar l'estat del món adequat a la resposta del detector d'olor segons les posicions en les quals es trobin per tal de realitzar la intersecció amb l'estat actual del món.
Arguments:

- X: Posició X del món.
- Y: Posició Y del món.
- S: Resposta del detector d'olor per a la posició X Y.
- NewLocs: Noves localitzacions del món per fer la intersecció.

2.4 intersectLocs:

Realitza la intersecció del primer paràmetre PrevLocs/MidLocs amb la del segon paràmetre NewLocs/RCLocs, aquests es combinen per tal d'obtenir les noves localitzacions que es guardaran al tercer paràmetre MidLocs/FinalLocs.
Arguments:

- PrevLocs/MidLocs: Localitzacions prèvies a la primera intersecció o localitzacions posteriors a la primera intersecció, en el primer cas obtindrem les localitzacions inicials per a cada volta en el segon cas obtindrem les localitzacions després d'haver realitzat la primera intersecció.
- NewLocs/RCLocs: NewLocs són les localitzacions obtingudes per isBarcenasAround i que realitzaran la intersecció amb PrevLocs, RCLocs són les localitzacions obtingudes per rajoyAndCospedal i que realitzaran la intersecció amb MidLocs.
- MidLocs/FinalLocs: Resultats d'haver realitzat les interseccions.

2.5 rajoyInfo:

Es realitza l'actualització de l'estat de Rajoy, la columna on s'ha trobat i la seva resposta, en cas que aquest s'hagi trobat en cas contrari es continuarà amb la que ja es tenia fins al moment. **Arguments:**

- PasY: Columna on s'ha trobat o no prèviament a Rajoy.
- PasM: Resposta prèvia de Rajoy en cas que s'hagi trobat en cas contrari -1.
- Y: Columna actual en la qual podem haver trobat a Rajoy o no.
- M: Resposta actual que ens pot haver donat Rajoy o en cas contrari -1.
- FutY: Actualització de la posició on hem trobat a Rajoy si l'hem trobat.
- FutM: Actualització de la resposta que ens ha donat si ens l'ha donat, en cas contrari -1.

2.6 cospedalInfo:

Es realitza l'actualització de l'estat de Cospedal, la seva resposta, en cas que aquesta s'hagi trobat, en cas contrari es continuarà amb el que ja es tenia fins aquell moment. **Arguments:**

- PasC: Resposta prèvia de Cospedal si s'havia trobat prèviament en cas contrari -1.
- C: Resposta actual que ens pot haver donat Cospedal en cas de trobar-la en cas contrari -1.
- FutC: Actualització de l'estat de la resposta de Cospedal en cas que hi hagi canvi, si encara no s'ha trobat, continuarà sent -1.

2.7 rajoyAndCospedal:

Donades les actualitzacions de les dades de Rajoy i Cospedal, s'obtenen les possibles localitzacions de Bàrcenas i amb les que posteriorment farem la intersecció amb les obtingudes pel detector d'olor per tal d'obtenir les noves localitzacions. **Arguments:**

- FutY: Posició actualitzada de la posició on hem trobat a Rajoy.
- FutM: Resposta actualitzada donada per Rajoy o -1.
- FutC: Resposta actualitzada donada per Cospedal o -1.
- RCLocs: Localitzacions possibles de Bàrcenas donada la informació rebuda.

2.8 writeWorld:

Aquesta funció només mostra per pantalla el món que se li passa per argument línia a línia. **Arguments:**

- World: Món amb les possibles localitzacions de Bàrcenas després de realitzar un pas que passarà a ser mostrat per pantalla.

3 Funcionament del codi en Prolog:

La primera crida que es farà serà a execSeqofSteps, aquesta, es crida recursivament per tal de recorre tota la llista de passos que es passen per arguments, a cada crida s'aniran actualitzant les variables PasY, PasM i PasC, així com FinalLocs, aquestes es passaran a updateSequenceOfSteps com s'ha explicat prèviament serà la que executarà tots els mètodes necessaris per a realitzar la cerca.

UpdateSequenceOfSteps:

1. isBarcenasAround:
   Aquí tenim tots els estats possibles per a cada situació que ens podem trobar al món amb les dimensions especificades a l'hora de fer la crida al programa de Python. Aquí es descarten les opcions depenent de la posició X, Y i la resposta que ens dóna el detector d'olor S. Així es guardarà com a resultat el món en el qual segons el detector d'olor hauria d'estar Bàrcenas.

2. intersectLocs:
   Funció proporcionada al fitxer testSimpleBarcenasLocation.pl, rebrà el món original passat a UpdateSequenceOfSteps i el món obtingut per isBarcenasAround per tal de realitzar la intersecció d'aquests dos obtenint el món de Bàrcenas amb les posicions actualitzades.

3. rajoyInfo:
   S'actualitza l'estat de Rajoy, la columna on s'ha trobat i la resposta obtinguda en cas que s'hagi trobat, si no es conservarà la resposta obtinguda prèviament o -1 en cas que no s'hagi trobat.

4. cospedalInfo: Es realitza la mateixa actualització realitzada per Rajoy però ara per Cospedal, en aquest cas no ens cal actualitzar la columna, ja que només ens importa la seva resposta sobre si Mariano menteix o no.

5. rajoyAndCospedal: Un cop actualitzada la informació sobre Rajoy i Cospedal procedim a obtenir el món depenent en la resposta d'aquests, en cas que no es tingui la resposta dels dos o falti la d'un s'obtindrà el món original.

6. IntersectLocs: Obtingut el món segons la resposta obtinguda per Rajoy i Cospedal podem procedir a realitzar la intersecció del món obtingut amb la resposta del detector d'olor i el món obtingut al pas anterior, d'aquesta manera obtenim un nou món actualitzat segons les respostes de totes les fonts d'informació possibles.

7. reverse: S'inverteix el món obtingut prèviament per tal d'imprimir-lo de manera correcta, aquesta funció és de Prolog.

8. write: S'imprimeix la informació del pas que s'ha dut a terme.

9. writeWorld: S'imprimeix el món en forma de taulell de NxN.

Com s'ha dit anteriorment aquest procés es repetirà fins que la llista de passos passats per arguments es quedi buida.

Nota: Per tal d'extreure els elements de la llista per l'índex es fa servir una funció del mòdul library(lists) anomenada nth0/3, aquesta consisteix de 3 arguments, el primer indica la posició de la qual es vol extreure, el segon la llista de la qual s'ha d'extreure l'element i la tercera on guardar l'element.

4 Tests:

Tal com es demana s'adjunta un programa en Python anomenat test.py, aquest es compon de 4 tests per al programa Prolog on s'executen diferents passos per a diferents programes findingBarcenas.py, aquests programes a més de diferents passos estan fets per a mides diferents de manera que es pot comprovar que funciona per a diferents mides, concretament es proven per a mons de mida $3x3$, $4x4$, $5x5$, $6x6$, al final de cada test es mostra el resultat final al qual s'hauria d'arribar en finalitzar l'execució de Prolog.

5 Informació Adicional:

Tant findingBarcenas.py com test.py s'ha programat en un:

```
Mac OS X Sierra 10.12.4 Darwin 16.5.0
Versió Python: 2.7.10
Versió Prolog: 7.2.3
```

Tant en findingBarcenas.py, test.py com el fitxer que es genera al executar el programa ha estat probat en un:

```
Mac OS X Sierra 10.12.4 Darwin 16.5.0
Fedora 25, 4.10.10-200.fc25.x86_64 GNU/Linux
```

6 Bibliografia:

http://stackoverflow.com/questions/12939425/prolog-access-specific-member-of-list
https://cv.udl.cat/access/content/group/102040-1617/labs/inference/cp1/testSimpleBarcenasLocation.pl
http://stackoverflow.com/questions/11525470/os-system-commands
https://docs.python.org/2/library/os.html