

Project Τεχνητής Νοημοσύνης

Γιάννης Οικονομίδης AM: 3711

Κων/νος Αραβανής AM: 3628

18 Νοεμβρίου 2008

1. Για την αναπαράσταση της κάθε κατάστασης του προβλήματος χρησιμοποιείται το κατηγορημα $location(X,Y)$ το οποίο δηλώνει την όχθη (Y: δεξιά, αριστερά) στην οποία βρίσκεται η κάθε οντότητα (X: αγρότης, λύκος, χόρτα, κατσίκα).
2. Για να απεικονίσουμε κάθε κατάσταση στην οθόνη δημιουργήσαμε έναν κανόνα ο οποίος εμφανίζει δύο στήλες, στην πρώτη εκ των οποίων εμφανίζονται οι οντότητες και στην δεύτερη εμφανίζεται η όχθη στην οποία βρίσκεται η κάθε μία από αυτές. Για παράδειγμα, αν υποθέσουμε ότι όλοι βρίσκονται στην δεξιά όχθη, έχουμε:

```
?- status.
```

```
-----STATUS-----  
PERSON          POSITION  
agrotis         deksia  
lykos           deksia  
katsika         deksia  
xorto           deksia  
-----
```

3. (α') Παρακάτω περιγράφουμε το κωδικά μας:
 - Αρχικά ορίσαμε στον κώδικα μας τα δυναμικά κατηγορήματα, ώστε να μπορούμε να αποθηκεύουμε και να διαγράφουμε από την μνήμη.
 - Ορίσαμε τις αποδεκτές οντότητες του προγράμματος μας.
 - Ορίσαμε τις αποδεκτές όχθες.
 - Ορίσαμε τις αρχικές καταστάσεις.
 - Δημιουργήσαμε έναν κανόνα ο οποίος έχει μόνο στόχο ώστε να εκτελείται αυτόματα καθορίζοντας τις τελικές καταστάσεις στις οποίες πρέπει να βρεθούν οι οντότητες μας.

- *other_side(X,Z)*: Ελέγχουμε αν η X είναι η αντίθετη πλευρά από αυτή στην οποία βρίσκεται ο Z.
- *approvable_lykos_katsika, approvable_xorto_katsika*: Κανόνες για την αποφυγή μη-αποδεκτής κατάστασης μετά από κάθε μετακίνηση.
- *boat_location(Z)*: Ελέγχει αν η βάρκα βρίσκεται στην όχθη Z.
- *update(Z,X)*: Ενημερώνει την μνήμη για την μετακίνηση καποιας οντότητας διαγράφοντας παράλληλα την προηγούμενη κατάσταση της.
- *quit*: Διαγράφει τις αποθηκευμένες καταστάσεις (τρέχοντες και τελικές).
- *win*: Κανόνας για τον έλεγχο επίτευξης τελικής κατάστασης.
- *move(X,Y)*: Κανόνας για την μετακίνηση της οντότητας X στην όχθη Y. Να σημειώσουμε ότι υπάρχουν διαφορετικοί τέτοιοι κανόνες για κάθε οντότητα, αφού για κάθε μία πρέπει να λάβουμε διαφορετικούς περιορισμούς.

Παρουσίαση κώδικα (βρίσκεται στο αρχείο *code_c1.pl*):

```
%Dilosi sinartiseon me dynamika katigorimata
:-dynamic location/2.
:-dynamic movable/1.
:-dynamic final_location/2.

%"Persons" pros metakinisi
movable(agrotis).
movable(lykos).
movable(katsika).
movable(xorto).

%Diatheimes pleures
side(deksia).
side(aristera).

location(agrotis,deksia).
location(katsika,deksia).
location(xorto,deksia).
location(lykos,deksia).

%Kathorismos telikon katastaseon
:- movable(X),location(X,Y),side(Z),\+Y==Z,assert(final_location(X,Z)),retract(movable(X)),fail.

%Arxiko minima pros ektuposi gia enimerosi tou paixti
:- nl,write('-----STARTING INFO-----'),nl,
  write('Person: katsika, agrotis, lykos, xorto'),nl,
  write('Side: deksia, aristera'),nl,
  write('-----FUNCTIONS-----'),nl,
  write('move(X,Y):'),nl,
  write('  Move X "person" to Y side'),nl,
  write('quit:'),nl,
  write('  Quit'),nl,
  write('-----'),nl.

%Elegxoume an o Z einai idi stin X pleula pou thelei na paei, kai an afti einai
%apodekti pleura
```

```

other_side(X,Z):-      (side(X),location(Z,Y),\+X==Y);
                      (write('-----ERROR-----'),nl,
                        write('The side tha you had typed is not\approvable'),nl,
                        write('-----'),nl,
                        !,fail).

%Apofugei ksemonaxiasmatos katsikas apo lyko.
approvable_lykos_katsika:-((location(lykos,aristera),location(katsika,deksia));
                           (location(lykos,deksia),location(katsika,aristera)));
                           (write('-----ERROR-----'),nl,
                             write('Lykos is going to kill katsika.'),nl,
                             write('-----'),nl,
                             !,fail).

%Apofugi mastouromatos katsikas.
approvable_xorto_katsika:- ((location(katsika,aristera),location(xorto,deksia));
                             (location(katsika,deksia),location(xorto,aristera)));
                             (write('-----ERROR-----'),nl,
                               write('Katsika is going to smoke xorto.'),nl,
                               write('-----'),nl,
                               !,fail).

%Des an i varka einai stin pleura X.
boat_location(Z):-      (location(Z,Y),location(agrotis,X),X==Y);
                      (write('-----ERROR-----'),nl,
                        write('There is no boat in this side.'),nl,
                        write('-----'),nl,
                        !,fail).

%Enimerosi neas katastasis. To Z einai to person kai to X to side sto opoio tha paei.
update(Z,X):-           retract(location(Z,_)),assert(location(Z,X)).

%Katharismos mnimis apo tis katastaseis pou exoume eisagei.
quit:-      retract(location(_,_)),retract(final_location(_,_)),fail.

2askisiinartisi elegxou nikis!
win:-      location(agrotis,X), final_location(agrotis,Y), X==Y,
            location(katsika,Z), final_location(katsika,W), Z==W,
            location(xorto,A), final_location(xorto,B), A=B,
            location(lykos,C), final_location(lykos,D), C==D,
            write('-----CONGRATS-----'),nl,
            write('          You ve won tha game!!!'),nl,
            write('-----'),nl,
            !,quit,fail.

%Oi move sinartiseis kounane to proto orisma (person) stin side X.

move(agrotis, X):-      other_side(X,agrotis),!,
                        approvable_lykos_katsika,!,
                        approvable_xorto_katsika,!,
                        update(agrotis,X),
                        !,win,fail.

move(lykos,X):-         other_side(X,lykos),!,
                        boat_location(lykos),!,
                        approvable_xorto_katsika,!,
                        update(lykos,X),
                        update(agrotis,X),
                        !,win,fail.

move(katsika,X):-      other_side(X,katsika),!,
                        boat_location(katsika),!,
                        update(katsika,X),
                        update(agrotis,X),
                        !, win, fail.

```

```

move(xorto,X):-
                                other_side(X,xorto),!,
                                boat_location(xorto),!,
                                approvable_lykos_katsika,!,
                                update(xorto,X),
                                update(agrotis,X),
                                !, win, fail.

```

(β') Στο ερώτημα αυτό της άσκησης απλά προσθέσαμε τον κανόνα *status* για την εμφάνιση στην οθόνη της κατάστασης της κάθε οντότητας. Έπίσης προσθέσαμε και την κλήση του κανόνα αυτού σε κάθε *move(X,Y)* ώστε να ενημερώνεται ο χρήστης αυτόματα για την νέα κατάσταση του προβλήματος.

Παρουσίαση κώδικα (βρίσκεται στο αρχείο *code_c2.pl*):

```

%Dilosi sinartiseon me dynamika katigorimata
:-dynamic location/2.
:-dynamic movable/1.
:-dynamic final_location/2.

%"Persons" pros metakinisi
movable(agrotis).
movable(lykos).
movable(katsika).
movable(xorto).

%Diathesimes pleures
side(deksia).
side(aristera).

location(agrotis,deksia).
location(katsika,deksia).
location(xorto,deksia).
location(lykos,deksia).

%Kathorismos telikon katastaseon
:- location(X,Y),movable(X),side(Z),\+Y==Z,assert(final_location(X,Z)),retract(movable(X)),fail.

%Arxiko minima pros ektuposi gia enimerosi tou paixti
:- nl,write('-----STARTING INFO-----'),nl,
  write('Person: katsika, agrotis, lykos, xorto'),nl,
  write('Side: deksia, aristera'),nl,
  write('-----FUNCTIONS-----'),nl,
  write('status:'),nl,
  write('  To see the current status'),nl,nl,
  write('move(X,Y):'),nl,
  write('  Move X "person" to Y side'),nl,
  write('quit:'),nl,
  write('  Quit'),nl,
  write('-----'),nl.

%Elegxoume an o Z einai idi stin X pleula pou thelei na paei, kai an afti einai
%apodekti pleura
other_side(X,Z):-
                                (side(X),location(Z,Y),\+X==Y);
                                (write('-----ERROR-----'),nl,
                                  write('The side tha you had typed is not\approvable'),nl,
                                  write('-----'),nl,
                                  !,fail).

%Apofugei ksemonaxiasmatos katsikas apo lyko.
approvable_lykos_katsika:-((location(lykos,aristera),location(katsika,deksia));
                             (location(lykos,deksia),location(katsika,aristera)));
                             (write('-----ERROR-----'),nl,

```

```

write('Lykos is going to kill katsika. '),nl,
write('-----'),nl,
!,fail).

%Apofugi mastouromatos katsikas.
approvable_xorto_katsika:- ((location(katsika,aristera),location(xorto,deksia));
                             (location(katsika,deksia),location(xorto,aristera)));
(write('-----ERROR-----'),nl,
 write('Katsika is going to smoke xorto. '),nl,
 write('-----'),nl,
 !,fail).

%Des an i varka einai stin pleura X.
boat_location(Z):- (location(Z,Y),location(agrotis,X),X==Y);
                   (write('-----ERROR-----'),nl,
                    write('There is no boat in this side. '),nl,
                    write('-----'),nl,
                    !,fail).

%Enimerosi neas katastasis. To Z einai to person kai to X to side sto opoio tha paei.
update(Z,X):- retract(location(Z,_)),assert(location(Z,X)).

%Katharismos mnimis apo tis katastaseis pou exoume eisagei.
quit:- retract(location(_,_)),retract(final_location(_,_)),fail.

2askisiinartisi elegxou nikis!
win:- location(agrotis,X), final_location(agrotis,Y), X==Y,
       location(katsika,Z), final_location(katsika,W), Z==W,
       location(xorto,A), final_location(xorto,B), A=B,
       location(lykos,C), final_location(lykos,D), C==D,
       write('-----CONGRATS-----'),nl,
       write('      You ve won tha game!!!'),nl,
       write('-----'),nl,
       !,quit,fail.

%Oi move sinartiseis kounane to proto orisma (person) stin side X.

move(agrotis, X):- other_side(X,agrotis),!,
                  approvable_lykos_katsika,!,
                  approvable_xorto_katsika,!,
                  update(agrotis,X),
                  status,!,win,fail.

move(lykos,X):- other_side(X,lykos),!,
                boat_location(lykos),!,
                approvable_xorto_katsika,!,
                update(lykos,X),
                update(agrotis,X),
                status,!,win,fail.

move(katsika,X):- other_side(X,katsika),!,
                  boat_location(katsika),!,
                  update(katsika,X),
                  update(agrotis,X),
                  status,!, win, fail.

move(xorto,X):- other_side(X,xorto),!,
                boat_location(xorto),!,
                approvable_lykos_katsika,!,
                update(xorto,X),
                update(agrotis,X),
                status,!, win, fail.

%Emfanisi katastaseon
status:- write('-----STATUS-----'),nl,

```

```

write('PERSON\t\tPOSITION'),nl,
location(X,Y),write(X),write('\t\t'),write(Y),nl,fail;
write('-----'),nl.

```

4. Στο σημείο αυτό τροποποιήσαμε τον κώδικα ώστε ο χρήστης να είναι αυτός που θα καθορίζει τις αρχικές καταστάσεις. Οι τροποποιήσεις που κάναμε είναι οι εξής:

- *start_locations_OK*: 'Ορίσαμε αυτό το επιπλέον δυναμικό κατηγορήμα ώστε μετά τον πρώτο επιτυχή έλεγχο του ότι ο χρήστης εισήγαγε όλες τις αρχικές καταστάσεις, να αποτρέψουμε τον επανέλεγχο.
- *is_movable(X)*: Ελέγχει αν ο *X* είναι αποδεκτή οντότητα. Σε αντίθετη περίπτωση εμφανίζει μήνυμα λάθους.
- *is_side(X)*: Όμοια, ελέγχει αν η *X* είναι αποδεκτή όχθη. Σε αντίθετη περίπτωση εμφανίζει μήνυμα λάθους.
- *start_location(X,Y)*: Γράφει στη μνήμη του προγράμματος την αρχική κατάσταση (όχθη *Y*) που όρισε ο χρήστης για την οντότητα *X*.
- *start_approvable*: Ελέγχει αν οι αρχικές καταστάσεις που όρισε ο χρήστης είναι για όλες τις οντότητες αποδεκτές με βάση τους περιορισμούς, αλλιώς του ζητά να τις ξαναορίσει.
- *check*: Ελέγχει αν έχουν δοθεί αρχικές καταστάσεις για όλες τις οντότητες και καλεί την *start_approvable*.
- Τέλος, τροποποιήσαμε την *move* ώστε να καλεί την *check* κάθε φορά που ο χρήστης δεν έχει δώσει πλήρεις-σωστές αρχικές καταστάσεις για τις οντότητες. Αυτό επιτυγχάνεται με τη βοήθεια του δυναμικού κατηγορήματος *start_locations_OK* (το οποίο θα έχει πάρει την τιμή 1).

Παρουσίαση κώδικα (βρίσκεται στο αρχείο *code.pl*):

```

%Dilosí sinartiseon me dynamika katigorimata
:-dynamic location/2.
:-dynamic movable/1.
:-dynamic final_location/2.
:-dynamic start_locations_OK/1.

%"Persons" pros metakinisi
movable(agrotis).
movable(lykos).
movable(katsika).
movable(xorto).

%Diathesimes pleures
side(deksia).
side(aristera).

%
start_locations_OK(0).

```

```

%Arxiko minima pros ektuposi gia enimerosi tou paixti
:- nl,write('-----STARTING INFO-----'),nl,
   write('First you must put every "person"'),nl,
   write('at the side that you want.'),nl,
   write('Person: katsika, agrotis, lykos, xorto'),nl,
   write('Side: deksia, aristera'),nl,
   write('-----FUNCTIONS-----'),nl,
   write('start_location(X,Y):'),nl,
   write('    The X person goes to the Y side.'),nl,
   write('    This function is used only for'),nl,
   write('    initialization'),nl,nl,
   write('status:'),nl,
   write('    To see the current status'),nl,nl,
   write('move(X,Y):'),nl,
   write('    Move X "person" to Y side'),nl,
   write('quit:'),nl,
   write('    Quit'),nl,
   write('-----'),nl.

%Des an to X einai person
is_movable(X):-      movable(X);
                    (write('-----ERROR-----'),nl,
                     write('This is not a "person"'),nl,
                     write('-----'),nl,
                     !,fail).

%Des an to X einai side
is_side(X):-         side(X);
                    (write('-----ERROR-----'),nl,
                     write('This is not a side'),nl,
                     write('-----'),nl,
                     !,fail).

%Elegxoume an o Z einai idi stin X pleula pou thelei na paei, kai an afti einai
%apodekti pleura
other_side(X,Z):-    (side(X),location(Z,Y),\+X==Y);
                    (write('-----ERROR-----'),nl,
                     write('The side tha you had typed is not\napprovable'),nl,
                     write('-----'),nl,
                     !,fail).

2askisiinartisi gia to kathorismo arxikon katastaseon apo ton xristi.
%To X einai to person kai to Y i side tou.
start_location(X,Y):- is_movable(X),!,
                     is_side(Y),!,
                     assert(location(X,Y)), other_side(Z,X), assert(final_location(X,Z)),
                     retract(movable(X)),!,fail.

%Elegxos gia to an i start locations einai apodektes
start_approvable:-  ((side(X),location(katsika,X),location(agrotis,X));
                    (side(Y),location(lykos,Y),location(agrotis,Y),side(W),
                     (\+location(katsika,W);\+location(xorto,W)));
                    (side(Z),location(lykos,Z),location(xorto,Z),side(A),
                     location(katsika,A),location(agrotis,A));
                    (side(B),location(xorto,B),location(agrotis,B),side(C),
                     (\+location(katsika,C);\+location(lykos,C))));
                    (write('-----ERROR-----'),nl,
                     write('Set start location again, because\nthere is conflict.'),nl,
                     write('-----'),nl,
                     !,quit,fail).

%Apofugei ksemonaxiasmatos katsikas apo lyko.
approvable_lykos_katsika:-((location(lykos,aristera),location(katsika,deksia));
                          (location(lykos,deksia),location(katsika,aristera)));

```

```

(write('-----ERROR-----'),nl,
write('Lykos is going to kill katsika.'),nl,
write('-----'),nl,
!,fail).

%Apofugi mastouromatos katsikas.
approvable_xorto_katsika:- ((location(katsika,aristera),location(xorto,deksia));
(location(katsika,deksia),location(xorto,aristera)));
(write('-----ERROR-----'),nl,
write('Katsika is going to smoke xorto.'),nl,
write('-----'),nl,
!,fail).

%Elegxei an exoun mpei ola ta dedomena.
check:-
((location(agrotis,_),
location(katsika,_),
location(xorto,_),
location(lykos,_));
(write('-----ERROR-----'),nl,
write('You ve missed someone.\nSet a start location for everyone'),
nl,write('-----'),nl,
!,fail)),!,start_approvable,retract(start_locations_OK(0)),
assert(start_locations_OK(1)).

%Des an i varka einai stin pleura X.
boat_location(Z):- (location(Z,Y),location(agrotis,X),X==Y);
(write('-----ERROR-----'),nl,
write('There is no boat in this side.'),nl,
write('-----'),nl,
!,fail).

%Enimerosi neas katastasis. To Z einai to person kai to X to side sto opoio tha paei.
update(Z,X):- retract(location(Z,_)),assert(location(Z,X)).

%Katharismos mnimis apo tis katastaseis pou exoume eisagei.
quit:- retract(location(_,_)),retract(final_location(_,_)),fail.

Zaskisiinartisi elegxou nikis!
win:-
location(agrotis,X), final_location(agrotis,Y), X==Y,
location(katsika,Z), final_location(katsika,W), Z==W,
location(xorto,A), final_location(xorto,B), A=B,
location(lykos,C), final_location(lykos,D), C=D,
write('-----CONGRATS-----'),nl,
write(' You ve won the game!!!'),nl,
write('-----'),nl,
!,quit,fail.

%Oi move sinartiseis kounane to proto orisma (person) stin side X.
move(agrotis, X):- (start_locations_OK(1);check),!,
other_side(X,agrotis),!,
approvable_lykos_katsika,!,
approvable_xorto_katsika,!,
update(agrotis,X),
status,!,win,fail.

move(lykos,X):- (start_locations_OK(1);check),!,
other_side(X,lykos),!,
boat_location(lykos),!,
approvable_xorto_katsika,!,
update(lykos,X),
update(agrotis,X),
status,!,win,fail.

move(katsika,X):- (start_locations_OK(1);check),!,

```



```

                                other_side(X,katsika),!,
                                boat_location(katsika),!,
                                update(katsika,X),
update(agrotis,X),                                status,!, win, fail.

move(xorto,X):-                                (start_locations_OK(1);check),!,
                                other_side(X,xorto),!,
                                boat_location(xorto),!,
                                approvable_lykos_katsika,!,
                                update(xorto,X),
                                update(agrotis,X),
                                status,!, win, fail.

%Emfanisi katastaseon
status:- write('-----STATUS-----'),nl,
        write('PERSON\t\tPOSITION'),nl,
        location(X,Y),write(X),write('\t\t'),write(Y),nl,fail;
        write('-----'),nl.

```

5. Στο σημείο αυτό μας ζητήθηκε να εκτελέσουμε το κώδικα ώστε να καταλήξουμε στην τελική κατάσταση. Τα βήματα που ακολουθήσαμε παρουσιάζονται παρακάτω (όπως τα μεταφέραμε από την κονσόλα):

```
?- [code].
```

```

-----STARTING INFO-----
First you must put every "person"
at the side that you want.
Person: katsika, agrotis, lykos, xorto
Side: deksia, aristera
-----FUNCTIONS-----
start_location(X,Y):
    The X person goes to the Y side.
    This function is used only for
    initialization

status:
    To see the current status

move(X,Y):
    Move X "person" to Y side
quit:
    Quit
-----
% code compiled 0.00 sec, 15,096 bytes
true.

?- start_location(katsika,aristera).

```

```

false.

?- start_location(agrotis,aristera).
false.

?- start_location(lykos,aristera).
false.

?- start_location(xorto,aristera).
false.

?- status.
-----STATUS-----
PERSON          POSITION
katsika          aristera
agrotis          aristera
lykos            aristera
xorto            aristera
-----
true.

?- move(katsika,deksia)
|      .
-----STATUS-----
PERSON          POSITION
lykos            aristera
xorto            aristera
katsika          deksia
agrotis          deksia
-----
false.

?- move(agrotis,aristera).
-----STATUS-----
PERSON          POSITION
lykos            aristera
xorto            aristera
katsika          deksia
agrotis          aristera
-----
false.

?- move(lykos,deksia).
-----STATUS-----
PERSON          POSITION

```

```

xorto          aristera
katsika        deksia
lykos          deksia
agrotis        deksia
-----
false.

?- move(katsika,aristera).
-----STATUS-----
PERSON        POSITION
xorto          aristera
lykos          deksia
katsika        aristera
agrotis        aristera
-----
false.

?- move(xorto,deksia).
-----STATUS-----
PERSON        POSITION
lykos          deksia
katsika        aristera
xorto          deksia
agrotis        deksia
-----
false.

?- move(agrotis,aristera).
-----STATUS-----
PERSON        POSITION
lykos          deksia
katsika        aristera
xorto          deksia
agrotis        aristera
-----
false.

?- move(katsika,deksia).
-----STATUS-----
PERSON        POSITION
lykos          deksia
xorto          deksia
katsika        deksia
agrotis        deksia
-----

```

```
-----CONGRATS-----  
      You ve won tha game!!!  
-----  
false.  
  
?-
```

Εργαλεία ανάπτυξης

Η συγγραφή και ανάπτυξη του κώδικα έγινε σε GNU/Linux, Debian Lenny Distro.

Ο κώδικας εκτελέστηκε με *swi-prolog* σε *J-Prolog IDE*.

Όσον αφορά την αναφορά και την παρουσίαση του κώδικα, χρησιμοποιήθηκε το **L^AT_EX**.