



ΠΑΝΕΠΙΣΤΗΜΙΟ  
ΠΑΤΡΩΝ  
UNIVERSITY OF PATRAS

---

## Τεχνητή Νοημοσύνη Ι

Εργαστηριακή Άσκηση 8

Σγάρμπας Κυριάκος

Τμήμα Ηλεκτρολόγων Μηχανικών και Τεχνολογίας Υπολογιστών

---

**ΑΝΟΙΚΤΑ** ακαδημαϊκά **ΠΠ**  
μαθήματα



## ΤΕΧΝΗΤΗ ΝΟΗΜΟΣΥΝΗ Ι - ΕΡΓΑΣΤΗΡΙΟ

### ΑΣΚΗΣΗ #8: “NIM και Τρίλιζα” (Αναζήτηση με Αντιπαλότητα)

1. Σε παιχνίδια με περιορισμένο μέγεθος του χώρου καταστάσεων, η ευθεία προσέγγιση είναι να τον εξερευνούμε ως εξής: Θεωρούμε ότι κάθε *θέση* (κατάσταση που περιέχει και την πληροφορία του ποιος έχει σειρά να παίξει) μπορεί να χαρακτηριστεί ως *κερδισμένη*, *χαμένη* ή *ισόπαλη*. Αν χαρακτηρίσουμε έτσι όλες τις θέσεις του χώρου καταστάσεων γνωρίζουμε πάντα ποιά είναι η καλύτερη κίνηση που έχουμε κάθε φορά στη διάθεσή μας. Ο παρακάτω κώδικας εξερευνά με αυτόν τον τρόπο οποιοδήποτε παιχνίδι μηδενικού αθροίσματος δύο παικτών και μπορεί να τρέξει πριν καν ακόμη ξεκινήσει το παιχνίδι για να δημιουργήσει μια βάση γνώσης. Μελετήστε τη λογική και τη λειτουργία του.

```
:- dynamic inKB/2.           % Ορισμός δυναμικής Βάσης Γνώσης (KB).
                             % inKB(+State,?Result)

% utility(+State,-Result)
% Δέχεται μια θέση/κατάσταση και επιστρέφει τον χαρακτηρισμό της: win,draw,lose
utility(State,X):- inKB(State,X), !. % Αν ο χαρακτηρισμός βρεθεί στην KB,
                             % ..χρησιμοποιείται απευθείας.

utility(State,X):-
    findall(                   % Αλλιώς βρίσκουμε όλες..
        (NextState,NextX),
        (successor(State,NextState), % ..τις διάδοχες καταστάσεις,
        utility(NextState,NextX)), % ..υπολογίζουμε αναδρομικά τα utilities τους
        NUTILSmult),          % ..και τα συγκεντρώνουμε σε μια λίστα πλειάδων.
    sort(NUTILSmult,NUTILS),  % Διαγράφουμε τυχόν διπλές πλειάδες.
    estimate(NUTILS,X),       % Υπολογίζουμε τον χαρακτηρισμό της αρχικής θέσης
    asserta(inKB(State,X)).   % ..και τον προσθέτουμε στην KB.

% estimate(+LIST,-Result)    % ΠΡΟΣΟΧΗ στο πρότυπο κλήσης!
% Αν υπάρχει έστω μία χαμένη θέση στη λίστα, η αρχική θέση είναι κερδισμένη.
% Αλλιώς, αν υπάρχει έστω και μία ισόπαλη θέση, η αρχική είναι ισόπαλη.
% Αλλιώς η αρχική είναι χαμένη.
estimate(NUTILS,win):- member( (_,lose),NUTILS), !.
estimate(NUTILS,draw):- member( (_,draw),NUTILS), !.
estimate( _,lose).
```

2. Στον παραπάνω κώδικα τόσο η **successor/2** όσο και η **inKB/2** εξαρτώνται από το παιχνίδι. Εδώ θα ασχοληθούμε με το παιχνίδι NIM:

III  
IIIII  
IIIIIII

Στο NIM έχουμε 3 ομάδες στοιχείων (συνήθως γραμμές, ξυλάκια, σπίρτα, κλπ) και ο κάθε

παίκτης όταν είναι η σειρά του πρέπει να αφαιρέσει τουλάχιστον ένα. Αν αφαιρέσει περισσότερα από ένα πρέπει να είναι όλα συνεχόμενα στην ίδια ομάδα. Στην παραλλαγή που θα εξετάσουμε εδώ επιτρέπεται να αφαιρέσει στοιχεία και από τη μέση κάποιας ομάδας, σπάζοντάς την σε δύο. Ο παίκτης που θα αναγκαστεί να πάρει το τελευταίο στοιχείο χάνει. Ο παρακάτω κώδικας περιγράφει το παιχνίδι:

```
% Κανόνες NIM
% Αρχική θέση: [3,5,7]

inKB([1],lose). % Ο παίκτης που σβήνει το τελευταίο στοιχείο χάνει.
inKB([],win).

% successor(+State,-NextState)
% Επιστρέφει μη ντετερμινιστικά όλες τις διάδοχες θέσεις από μια αρχική.
successor([],[]):- !, fail. % Χωρίς στοιχεία το παιχνίδι τελειώνει.
successor([N],[Q]):- % Ο παίκτης μπορεί να σβήσει..
    NS is N-1, enlist(NS,AllNS), member(Q,AllNS).% ..ένα ή περισσότερα στοιχεία,
successor([_],[_]). % ..ή και ολόκληρη ομάδα.
successor([N],[L,R]):- % Μια ομάδα μπορεί να σπάσει σε δύο
    N>2, % ..αν έχει πάνω από 2 στοιχεία.
    NReduced is N-2, % Η κάθε νέα ομάδα θα έχει το πολύ N-2 στοιχεία
    % ..(1 που αφαιρέσαμε +1 min για την άλλη ομάδα)

    enlist(NReduced, LN),
    member(L, LN), member(M, LN), member(R, LN), % Όλες οι δυνατές τιμές των L,M,R
    L=<R, N =:= L+M+R. % ..φιλτράρονται από τις συνθήκες.

successor(Lin,Lout):-
    length(Lin,LEN), LEN>1, % Αν η θέση έχει πάνω από 1 ομάδα,
    remove(X,Lin,LmX), % ..επιλέγουμε μία,
    successor([X],SX), % ..βρίσκουμε την διάδοχη ομάδα,
    append(LmX,SX,LoutNS), % ..και την αντικαθιστούμε στη θέση.
    msort(LoutNS,Lout). % Ταξινομούμε τις ομάδες.

remove(X,Lin,Lout):- append(A,[X|B],Lin), append(A,B,Lout).

% enlist(+Integer,-List)
% ?- enlist(4,L)
% L=[4,3,2,1]
enlist(0,[]):- !.
enlist(X,[X|T]):- Y is X-1, enlist(Y,T).
```

Τρέξτε τον και εξετάστε αν καθεμία από τις θέσεις: [2,2,2], [3,3], [3,5,7], [2,4,6] [2,4,6,8] είναι κερδισμένη ή χαμένη. Κάθε φορά ελέγχετε με την εντολή **listing(inKB)** τις θέσεις που έχει προσθέσει η Prolog στη βάση γνώσης της. Αν θέλετε να σβήσετε τη βάση γνώσης χρησιμοποιήστε την εντολή **retractall(inKB(\_,\_))** και μετά κάντε **reconsult** του κώδικα.

3. Με δημιουργημένη τη βάση γνώσης η Prolog πλέον μπορεί να βρει την βέλτιστη κίνηση για κάθε θέση που θα της δώσουμε. Φτιάξτε μια συνάρτηση **play(+State,-NewState)** που θα χρησιμοποιεί την πληροφορία από τη βάση γνώσης και θα προτείνει την βέλτιστη κίνηση, όπως παρακάτω:

```
?- play([2,4,6],NewState).
I play from [2,4,6] to [1,4,6]
If you play perfectly, you will eventually win
NewState = [1, 4, 6].

?- play([3,5,7],NewState).
I play from [3,5,7] to [2,5,7]
If you play perfectly, you will eventually lose
NewState = [2, 5, 7].
```

Σημείωση: Ως βέλτιστη θεωρούμε την κίνηση που κερδίζει (αν υπάρχει αυτή η δυνατότητα),

χωρίς να μας ενδιαφέρει το πλήθος των κινήσεων μέχρι τη νίκη.

4. Ως *τέλεια παρτίδα* (*perfect game*) χαρακτηρίζεται μια ακολουθία κινήσεων από την αρχική κατάσταση μέχρι το τέλος του παιχνιδιού, στην οποία κανένας παίκτης δεν έκανε λάθος. Φτιάξτε μια συνάρτηση **perfect\_game(+State)** που θα ξεκινά από την κατάσταση *State* και θα τυπώνει όλες τις κινήσεις μιας τέλει παρτίδας μαζί με τους χαρακτηρισμούς τους, μέχρι το τέλος του παιχνιδιού. Πχ.:

```
?- perfect_game([3,5,7]).
[3,5,7] --> [2,5,7]   lose
[2,5,7] --> [1,5,7]   win
[1,5,7] --> [1,4,5]   lose
[1,4,5] --> [4,5]     win
[4,5] --> [4,4]     lose
[4,4] --> [3,4]     win
[3,4] --> [3,3]     lose
[3,3] --> [2,3]     win
[2,3] --> [2,2]     lose
[2,2] --> [1,2]     win
[1,2] --> [1]       lose
[1] --> []         win
```

5. Αλλάξτε την **successor/2** και την **inKB/2** ώστε το πρόγραμμα να παίζει Τρίλιζα.

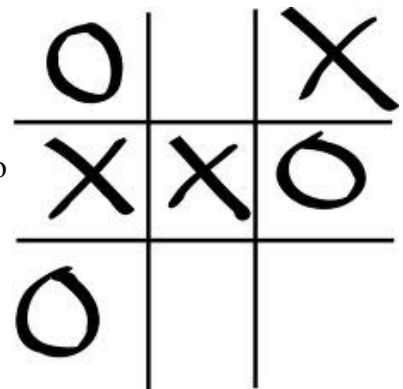
Σημείωση: Αντίθετα με το NIM, στην Τρίλιζα το παιχνίδι μπορεί να καταλήξει και σε ισοπαλία.

Υπόδειξη: Έχετε την ελευθερία να χρησιμοποιήσετε όποιον τρόπο θέλετε για την κωδικοποίηση των καταστάσεων, όμως ίσως σας διευκολύνει κάτι τέτοιο:

$([-1, 0, 1, 1, 1, -1, -1, 0, 0], 1) \rightarrow$  η θέση του

διπλανού σχήματος, εκφρασμένη κατά γραμμές, όπου κάθε 1 αντιστοιχεί σε X, κάθε -1 σε O και κάθε 0 σε κενό τετράγωνο. Το

τελευταίο 1 στην πλειάδα δηλώνει ποιός παίκτης έχει σειρά να παίξει. Αυτό θα μπορούσε να το συμπεράνει κάποιος και από το πλήθος των X και O, αλλά έτσι είναι απλοποιείται ο κώδικας.



6. Ελέγξτε τη σωστή λειτουργία των συναρτήσεων **utility/2**, **play/2** και **perfect\_game/1** στην Τρίλιζα.

# Σημειώματα

## Σημείωμα Ιστορικού Εκδόσεων Έργου

Το παρόν έργο αποτελεί την έκδοση **1.0**

- Έκδοση **1.0** διαθέσιμη [εδώ](#).

## Σημείωμα Αναφοράς

Copyright Πανεπιστήμιο Πατρων, Φακωτάκης Νικόλαος, Σγάρμπας Κυριάκος, Πέππας Παύλος, Μουστάκας Κωνσταντίνος. «Εργαστήριο 8». Έκδοση: 1.0. Πάτρα 2015. Διαθέσιμο από τη δικτυακή διεύθυνση: [https://eclass.upatras.gr/modules/course\\_metadata/opencourses.php?fc=15](https://eclass.upatras.gr/modules/course_metadata/opencourses.php?fc=15)

## Σημείωμα Αδειοδότησης

Το παρόν υλικό διατίθεται με τους όρους της άδειας χρήσης Creative Commons Αναφορά, Μη Εμπορική Χρήση Παρόμοια Διανομή 4.0 [1] ή μεταγενέστερη, Διεθνής Έκδοση. Εξαιρούνται τα αυτοτελή έργα τρίτων π.χ. φωτογραφίες, διαγράμματα κ.λ.π., τα οποία εμπεριέχονται σε αυτό και τα οποία αναφέρονται μαζί με τους όρους χρήσης τους στο «Σημείωμα Χρήσης Έργων Τρίτων».



[1] <http://creativecommons.org/licenses/by-nc-sa/4.0/>

Ως **Μη Εμπορική** ορίζεται η χρήση:

- που δεν περιλαμβάνει άμεσο ή έμμεσο οικονομικό όφελος από την χρήση του έργου, για το διανομέα του έργου και αδειοδόχο
- που δεν περιλαμβάνει οικονομική συναλλαγή ως προϋπόθεση για τη χρήση ή πρόσβαση στο έργο
- που δεν προσπορίζει στο διανομέα του έργου και αδειοδόχο έμμεσο οικονομικό όφελος (π.χ. διαφημίσεις) από την προβολή του έργου σε διαδικτυακό τόπο

Ο δικαιούχος μπορεί να παρέχει στον αδειοδόχο ξεχωριστή άδεια να χρησιμοποιεί το έργο για εμπορική χρήση, εφόσον αυτό του ζητηθεί.

## **Διατήρηση Σημειωμάτων**

- Οποιαδήποτε αναπαραγωγή ή διασκευή του υλικού θα πρέπει να συμπεριλαμβάνει:
- το Σημείωμα Αναφοράς
- το Σημείωμα Αδειοδότησης
- τη δήλωση Διατήρησης Σημειωμάτων
- το Σημείωμα Χρήσης Έργων Τρίτων (εφόσον υπάρχει)

μαζί με τους συνοδευόμενους υπερσυνδέσμους.

## **Σημείωμα Χρήσης Έργων Τρίτων**

Το Έργο αυτό κάνει χρήση των ακόλουθων έργων:

## Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στο πλαίσιο του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «**Ανοικτά Ακαδημαϊκά Μαθήματα στο Πανεπιστήμιο Αθηνών**» έχει χρηματοδοτήσει μόνο τη αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.

