

**Τμήμα:** Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών  
**Μάθημα:** Εργαστήριο Τεχνητής Νοημοσύνης.  
**Διδάσκοντες:** Καθηγ. Μανόλης Μαρακάκης,  
Δρ, Χαρίδημος Κονδυλάκης  
**Ημερομηνία παράδοσης** 4 Ιουνίου 2022  
Εαρινό εξάμηνο 2021-22

## Εργασία 5

Θεωρήσατε το παρακάτω Prolog πρόγραμμα. Το κατηγορημα `monopati(X, Y, Monopati)` είναι αληθές εάν `Monopati` είναι το μονοπάτι μεταξύ κορυφών `X` και `Y` σ' ένα κυκλικό κατευθυνόμενο γράφο.

```
monopati(X, Y, Monopati) :-  
    monopati1(X, Y, [X], Monopati).  
  
monopati1(X, X, Monopati, Monopati).  
monopati1(X, Z, Monopati, Teliko_monopati) :-  
    akme(X, Y),  
    \+ member(Y, Monopati),  
    append(Monopati, [Y], Neo_monopati),  
    monopati1(Y, Z, Neo_monopati, Teliko_monopati).
```

```
member(X,[H|T]) :- X=H.  
member(X,[H|T]) :- X \= H, member(X,T).
```

```
append([],L2,L2).  
append([H1|T1], L2, [H1|T3]) :- append(T1, L2, T3).
```

```
akme(a, c).  
akme(c, a).  
akme(c, e).  
akme(c, f).  
akme(b, c).  
akme(b, d).  
akme(d, f).  
akme(d, g).
```

Να αναπαραστήσετε το παραπάνω πρόγραμμα σε μη-βασικούς όρους. Κάθε πρόταση του προγράμματος σας θα είναι μια πρόταση της Prolog στη εξής μορφή.

`clause2(ΚεφαλήΠροτασης, ΣώμαΠρότασης)`

Για παράδειγμα, οι προτάσεις

```
monopati(X, Y, Monopati) :-  
    monopati1(X, Y, [X], Monopati).  
akme(a, c).
```

Θα παριστάνονται σε μη βασική μορφή αντίστοιχα ως εξής:

```
clause2(monopati(X, Y, Monopati), (monopati1(X, Y, [X], Monopati)).
```

clause2(akme(a, c), true).

Να αναπαραστήσετε την άρνηση «\+» ως νέο τελεστή, π.χ. «not», ως εξής «:-op(900, fy, [not]).».

Να γράψετε ένα μεταπρόγραμμα το οποίο θα κάνει τα εξής:

Αρχικά θα δίνει στο χρήστη τον εξής κατάλογο επιλογών:

1. Για ενημέρωση του προγράμματος-αντικείμενο.
2. Για κλήση του μετα-διερμηνέα solve/1 για να τρέξει κάποιους στόχους του προγράμματος-αντικείμενο, π.χ. monopati/3, τους οποίους θα δίνει ο χρήστης.
3. Για έξοδο από το μεταπρόγραμμα.

Εάν ο χρήστης επιλέξει 1 να μπαίνει στον εξής κατάλογο επιλογών.

- a. Για αλλαγής κάποιας πρότασης του προγράμματος.
- b. Για εισαγωγή κάποιας νέας πρότασης στο πρόγραμμα-αντικείμενο.
- c. Για διαγραφή κάποιας πρότασης από το πρόγραμμα-αντικείμενο.
- d. Για έξοδο από τη διαδικασία ενημέρωσης τους προγράμματος αντικείμενο.

Εάν ο χρήστης επιλέξει 2 θα μπαίνει στον εξής κατάλογο επιλογών.

- a. Για εισαγωγή στόχου τον οποίο θα εκτελεί ο μετα-διερμηνέας solve/1. Ο χρήστης θα δίνει το στόχο μέσω της διεπικοινωνίας του μεταπρογράμματος σας και το μεταπρόγραμμα θα καλεί τον μετα-διερμηνέα solve/1 ο οποίος θα εκτελεί το στόχο.
- b. Για έξοδο από τη διαδικασία εκτέλεσης στόχων του προγράμματος-αντικείμενο.

Για μετα-διερμηνέα να χρησιμοποιήσετε τον μετα-διερμηνέα, Πρόγραμμα 9.19, και εφόσον χρειάζεται να τον τροποποιήσετε ή να τον επεκτείνετε.

```
solve(true) :- !.  
solve( (A, B) ) :- !,  
    solve(A),  
    solve(B).  
  
solve(Goal) :-  
    \+ predicate_property (Goal, built_in),  
    clause2( Goal, Body ),  
    solve(Body).  
  
solve(Goal) :-  
    predicate_property (Goal, built_in),  
    call(Goal).
```

Να εξετάσετε το μεταπρόγραμμα σας για τα εξής δεδομένα.

1. Να αντικαταστήσετε την πρόταση «monopati1(X, X, Monopati, Monopati).» με την ισοδύναμη πρόταση «monopati1(X, X, Monopati, Teliko\_monopati) :- Monopati = Teliko\_monopati”.
2. Να διαγράψετε την πρόταση «akme(c, a).».
3. Να καταχωρήσετε την πρόταση «akme(e, a).»
4. Να τρέξετε το πρόγραμμα αντικείμενο για τους εξής στόχους «α. ?- monopati(a,f,Path). β. ?- monopati(a,Y,Path). γ. ?- monopati(b,f,Path). δ. ?- monopati(b,Y,Path).»