

ΠΛΗ31

PROLOG

Μάθημα 5: Έλεγχος Οπισθοδρόμησης

Δημήτρης Ψούνης



ΠΕΡΙΕΧΟΜΕΝΑ

A.Θεωρία

1. Οπισθοδρόμηση

1. Μη προσδοκώμενα αποτελέσματα

2. Το κατηγορημα ! /ο

2. Βίαιος Τερματισμός Προγράμματος

1. Το κατηγορημα fail / ο

B.Ασκήσεις

A. Θεωρία

1. Οπισθοδρόμηση

1. Μη προσδοκώμενα Αποτελέσματα

➤ Το παρακάτω πρόγραμμα εντοπίζει τον μέγιστο μεταξύ δύο αριθμών:

```
max(X, Y, M) :-  
    X > Y,  
    M = X.  
max(X, Y, M) :-  
    X <= Y,  
    M = Y.
```

➤ Εκτελούμε τις παρακάτω ερωτήσεις:

```
?- max(4, 5, N) .  
N = 5.
```

```
?- max(5, 4, N) .  
N = 5 ;  
false.
```

➤ Η δεύτερη απάντηση κρίνεται παράλογη και καταχρηστική!

X > Y,

M = X.

max(X, Y, M) :-

X <= Y,

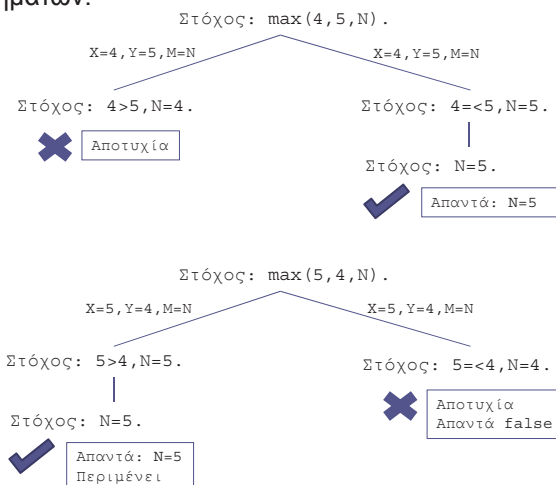
M = Y.

A. Θεωρία

1. Οπισθοδρόμηση

1. Μη προσδοκώμενα Αποτελέσματα

Για να δούμε γιατί συμβαίνει αυτό θα μελετήσουμε το δένδρο εκτέλεσης των δύο ερωτημάτων:



?- max(4, 5, N) .
N = 5.

?- max(5, 4, N) .
N = 5 ;
false.

A. Θεωρία

1. Οπισθοδρόμηση

2. Το κατηγορήμα ! / 0

- Το κατηγορήμα ! / 0 (διαβάζεται cut) χρησιμοποιείται για τον έλεγχο της οπισθοδρόμησης:
 - Η εκτέλεση του επιτυγχάνει πάντα.
 - Θέτει φραγμό στην οπισθοδρόμηση. Όταν συναντίεται κατά την οπισθοδρόμηση, αυτή σταματά!
- Το χρησιμοποιούμε όταν γνωρίζουμε ότι εφόσον επιτυγχάνει ένας δρόμος υπολογισμού δεν πρέπει να γίνει οπισθοδρόμηση από το σημείο χρήσης του ! και πριν.
- Διορθώνουμε τον ορισμό του κατηγορήματος max ως εξής:

```
max(X, Y, M) :-
    X > Y, !,
    M = X.
max(X, Y, M) :-
    X <= Y,
    M = Y.
```

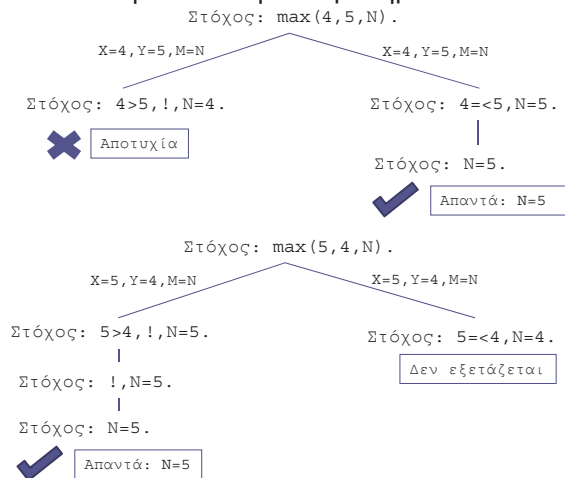
```
max(X, Y, M) :-
    X > Y, !,
    M = X.
max(X, Y, M) :-
    X <= Y,
    M = Y.
```

A. Θεωρία

1. Οπισθοδρόμηση

2. Το κατηγορήμα ! / 0

Βλέπουμε εκ νέου την εκτέλεση των ερωτημάτων



?- max(4, 5, N).
N = 5.

?- max(5, 4, N).
N = 5.

A. Θεωρία

1. Οπισθοδρόμηση

2. Το κατηγορήμα ! / 0

- Με χρήση του ! μπορούμε να έχουμε κομψές αναπαραστάσεις στα προγράμματα μας.
 - Στην ουσία προσομοιώνουμε και το if-else
- Καλύτερες αναπαραστάσεις του ίδιου κατηγορήματος:

```
max(X, Y, M) :-
    X > Y, !,
    M = X.
max(X, Y, M) :-
    X <= Y,
    M = Y.
```

```
max(X, Y, M) :-
    X > Y, !,
    M = X.
max(X, Y, M) :-
    M = Y.
```

```
max(X, Y, X) :-
    X > Y, !.
max(X, Y, Y).
```

A. Θεωρία

2. Βιαίος Τερματισμός

1. Το κατηγορήμα fail / 0

- Το κατηγορήμα fail / 0 είναι το κατηγορήμα της αποτυχίας:
 - Η εκτέλεση του αποτυγχάνει πάντα.
 - (άρα εκκινά την οπισθοδρόμηση)
- Από μόνο του δεν κάνει και πολλά πράγματα.
 - Ωστόσο αν έχουμε ακολουθιακά τις δηλώσεις **! και έπειτα fail** τερματίζουμε στην ουσία επι τόπου το πρόγραμμα.
 - Είναι χρήσιμο εργαλείο σε μεγάλες αναζητήσεις όταν καταλαβαίνουμε ότι πρέπει να τερματίσουμε την εκτέλεση του προγράμματος με αρνητική απάντηση.



Β. Ασκήσεις

Εφαρμογή 1

Έστω το παρακάτω πρόγραμμα Prolog:

```
first:- room(X),
        write(X),
        nl,
        fail.
second:- room(X),
        !,
        write(X),
        nl,
        fail.
third:- room(X),
        write(X),
        nl,
        !,
        fail.
fourth:- !,
        room(X),
        write(X),
        nl,
        fail.
fifth:- room(X),
        write(X),
        nl,
        fail,
        !.

room(a).
room(b).
room(c).
room(d).
```

Σας δίνονται τα παρακάτω ερωτήματα:

- (a1) first.
(a2) second.
(a3) third.
(a4) fourth.
(a5) fifth.

Για κάθε ένα από τα παραπάνω ερωτήματα αντιστοιχείστε μία από τις παρακάτω απαντήσεις:

(1)	(2)	(3)	(4)	(5)
a	a	a	a	a
No	b	No	b	b
	c		c	c
	d		d	d
	No		No	No

Γράψτε την απάντηση σε μορφή ζεύγους, π.χ. a3->2



Β. Ασκήσεις

Εφαρμογή 1



Β. Ασκήσεις

Εφαρμογή 2

Επιλέξτε από τις παρακάτω εναλλακτικές τον ορθό κώδικα Prolog, που υλοποιεί τη συνάρτηση κατωφλίου:

$$\varphi(v) = \begin{cases} 1, & v \geq 0 \\ 0, & v < 0 \end{cases}$$

Τεκμηριώστε την επιλογή σας παραθέτοντας τα αποτελέσματα για τις κλήσεις f(-1), f(5).

(α)	f(X,Y):- X<0,!, Y=0.
(β)	f(X,Y):- X<0,!, Y=0. f(,0).
(γ)	f(X,Y):- X>=0,!, Y=0. f(,1).
(δ)	f(X,Y):- X<0,!, Y=0. f(,1).
(ε)	f(,1). f(X,Y):- X<0,!, Y=0.



Β. Ασκήσεις

Εφαρμογή 2