ΠΛΗ31

PROLOG

Μάθημα 4:
Τελεστές και Αριθμητική σε Prolog

Δημήτρης Ψούνης

ΠΕΡΙΕΧΟΜΕΝΑ

Α. Θεωρία
1. Τελεστές
1. Αριθμητικοί Τελεστές

- Στην Prolog ορίζονται οι αριθμητικοί τελεστές:

Σύμβολο	Ερμηνεία
+	Πρόσθεση
−	Αφαίρεση
*	Πολλαπλασιασμός
/	Διαίρεση
//	Modulo

- Έχουμε δικαίωμα να γράψουμε οποιαδήποτε παράσταση χρησιμοποιώντας τους παραπάνω τελεστές, π.χ. 4+5*3-2 που εφαρμόζοντας την γνωστή προτεραιότητα των τελεστών (πρώτα πολ/μος, διαίρεση και mod έπειτα πρόσθεση και αφαίρεση) μεταφράζεται από την Prolog σε προθεματική μορφή ως +(4,-(*(5,3),2)) όπου οι τελεστές είναι συναρτησιακά σύμβολα

Α. Θεωρία
1. Τελεστές
2. Σχεσιακοί Τελεστές

- Στην Prolog ορίζονται οι σχεσιακοί τελεστές:

Σύμβολο	Ερμηνεία
>	Μεγαλύτερο από
<	Μικρότερο από
>=	Μεγαλύτερο ή ίσο
<=	Μικρότερο ή ίσο
=:=	Ίσον
\=	Διάφορον

- Οι τελεστές αυτοί εφαρμόζονται για τη σύγκριση <u>αριθμών</u>. Σε αντίθεση με το A=B και το \+(A=B) που αντίστοιχα ελέγχουν αν οι μεταβλητές A και B έχουν ενοποιηθεί με τις ίδιες σταθερές.

Α. Θεωρία
1. Τελεστές
3. Λογικοί Τελεστές

- Έχουμε ήδη μελετήσει τους λογικούς τελεστές:

Σύμβολο	Ερμηνεία
,	Λογικό AND
;	Λογικό OR
\+	Λογικό NOT

- Τέλος ο τελεστής =, αποκαλείται τελεστής ενοποίησης και αληθεύει αν οι συμβολοσειρές που έχουν αριστερά και δεξιά έχουν την ίδια τιμή.

Α. Θεωρία
1. Τελεστές
4. Ο τελεστής is

- Ένας ειδικός τελεστής είναι ο is. Είναι ο τελεστής που δίνει εντολή στην Prolog να υπολογίσει μια αριθμητική ποσότητα:

```
?- X=2*3
X=2*3.

?- X is 2*3.
X=6
```

  - Παρατηρήστε παραπάνω:
    - Το = ενοποιεί τον όρο στα δεξιά με την μεταβλητή X
    - Το is υπολογίζει τον όρο στα δεξιά και αναθέτει την τιμή στην μεταβλητή X

Α. Θεωρία
1. Τελεστές
4. Ο τελεστής is

- Ο ορισμός του τελεστή is μας επιτρέπει να κατασκευάσουμε κατηγορήματα που κάνουν υπολογισμό ποσοτήτων. Π.χ. ο ορισμός του κατηγορήματος length είναι:

```
length([],0).

length([X|L],N):-
   length(L,M),
   N is M+1.
```

  - (Άσκηση 1) Κατασκευάστε το δένδρο εκτέλεσης του ερωτήματος:

```
? - length([1,2],N).
```

  - (Ασκηση 2) Κατασκευάστε το κατηγόρημα sumlist/2 να αληθεύει αν το άθροισμα των στοιχείων της λίστας που δέχεται ως πρώτο όρισμα να ισούται με την μεταβλητή του 2^{ου} ορίσματος.

Β.Ασκήσεις
Εφαρμογή 1

Θέλουμε να βρούμε με πόσους τρόπους μπορούμε να «χαλάσουμε» 1 Ευρώ σε ψιλά. Έχουμε γράψει το ακόλουθο πρόγραμμα σε Prolog και μας έχουν διαβεβαιώσει πως όχι μόνο τρέχει, αλλά κάνει και πολλά παραπάνω απ' όσα θέλουμε, απλά δε μπορεί να χειριστεί πεντάλεπτα και δίλεπτα.

```
change([C50,C20,C10,P]) :-
    member(C50,[0,1,2]),                        /* 50 λεπτά */
    member(C20,[0,1,2,3,4,5]),                  /* 20 λεπτά */
    member(C10,[0,1,2,3,4,5,6,7,8,9,10]) ,      /* 10 λεπτά */
    S is 50*C50 + 20*C20 +10*C10,
    S =< 100,
    P is 100-S.
```

(α) Τι θ' απαντήσει το πρόγραμμα στο παρακάτω ερώτημα και γιατί;
`?- change([0,2,2,P]).`
(β) Τι θ' απαντήσει το πρόγραμμα στο παρακάτω ερώτημα και γιατί;
`?- change([0,X,5,P]).`
(γ) Αλλάξτε το παραπάνω πρόγραμμα ώστε να «χαλάει» οτιδήποτε κάτω του 1 Ευρώ σε ψιλά. Μην ασχοληθείτε με έλεγχο της εισόδου.

Β.Ασκήσεις
Εφαρμογή 2

Ορίστε το κατηγόρημα max/3, το οποίο όταν δέχεται 3 ορίσματα, έστω X,Y και Z να αληθεύει αν το Z έχει την ίδια τιμή με τον μεγαλύτερο από τους X,Y

Β.Ασκήσεις
Εφαρμογή 3

Ορίστε το κατηγόρημα prod/2. Το prod(L,P) επιστρέφει στο P το γινόμενο των αριθμών που είναι στοιχεία της λίστας L

Β.Ασκήσεις
Εφαρμογή 4

Ορίστε το κατηγόρημα factorial/2. Το factorial(N,F) επιστρέφει στο F το παραγοντικό του ακεραίου N

Β.Ασκήσεις
Εφαρμογή 5

Ορίστε το κατηγόρημα maxlist/2. Το maxlist(L,M) επιστρέφει στο M τον μεγιστο ακέραιο της λίστας L

Β.Ασκήσεις
Εφαρμογή 6

Ορίστε το κατηγόρημα first_n/3. Το first_n(N,L1,L2) επιστρέφει στην λίστα L2 τα πρώτα N στοιχεία της λίστας L1