

Τελεστές (Operators)

$$\begin{array}{c} \exists \alpha \Phi(\alpha) \\ \vdash \frac{\frac{x}{B(x)} \quad \frac{x}{A(x)}}{\neg K} \quad \forall x \\ P \rightarrow Q \end{array}$$

Δομή

- Τελεστές
- Ορισμός
 - Προτεραιότητα
 - Προσεταιριστικότητα
- Παράδειγμα

Σημειογραφία της Prolog

- Σε αρκετές περιπτώσεις η συνηθισμένη σημειογραφία (notation) της Prolog μπορεί να αποδειχθεί λίγο αδόκιμη/δυσνόητη.

- Συνηθισμένη αναπαράσταση μαθηματικών εκφράσεων:

$$32*X+44*Y-12$$

- Μεταφρασμένη σε μορφή όρων στην Prolog:

$$(+(* (32, X), *(44, Y)), 12)$$

- Ορθή, συνεπής αλλά όχι πρόσφορη.

Τελεστές

- Η σημειογραφία της Prolog επεκτείνεται ώστε να υποστηρίζει τη συνήθη ενθεματική μορφή μέσω του ορισμού **τελεστών**.
- Η Prolog έχει ήδη ένα πλήθος τελεστών που ορίζονται.
 - “Βέλος” (':-'), το κόμα (',') και αριθμητικούθς τελεστές.

Τελεστές (cont)

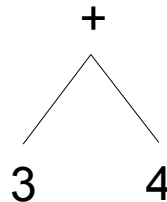
- Οι τελεστές δεν μεταβάλλουν την λειτουργική σημασιολογία της γλώσσας.
 - Syntactic sugar
 - Εσωτερικά τα πάντα μεταφράζονται στην συνηθισμένη μορφή όρων της Prolog.

?- X is 3+4.

X = 7

?- X is +(3,4)

X = 7.



Ενθεματική μορφή

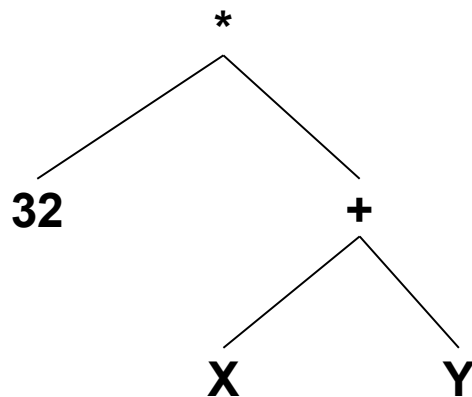
- Η σαφής μετάφραση μιας παράστασης η οποία περιέχει ενθεματικούς τελεστές απαιτεί τον ορισμό δύο πραγμάτων:
 - Προτεραιότητα (Precedence)
 - Προσεταιριστικότητα (Associativity)

Προτεραιότητα

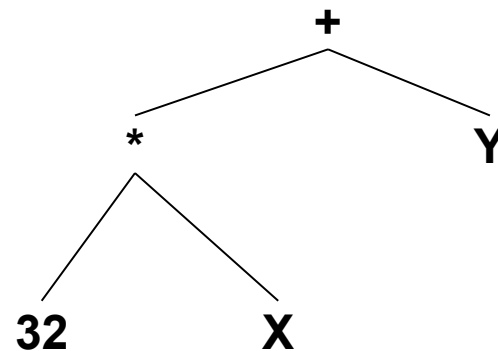
- Στη έκφραση

$32 * X + Y$

- ποια η σωστή μετάφραση;



(a) $*(32,+(X,Y))$



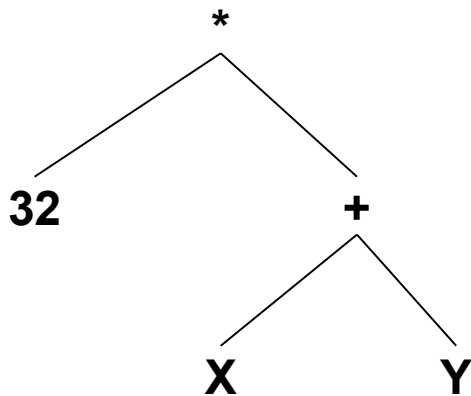
(b) $+(*(32,X),Y)$

Προτεραιότητα Τελεστών

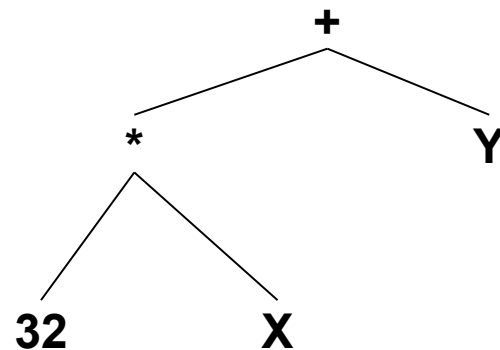
- Αριθμητική τιμή από 0 to 1200
 - (στις περισσότερες υλοποιήσεις της Prolog)
 - Εκφράζει **σειρά (order)**
- Καθορίζει ποιος τελεστής είναι το κύριο συναρτησιακό σύμβολο (functor) μιας έκφρασης.
- **Μεγάλη τιμή, μικρή προτεραιότητα.** Άρα ο τελεστής με την *μεγαλύτερη τιμή αποτελεί το κύριο συναρτησιακό σύμβολο (root)*.

Μετάφραση με βάση την προτεραιότητα

- Αν ο τελεστής "*" έχει μεγαλύτερη **τιμή σειράς** σε σχέση με το "+" τότε το σωστό είναι το (a);
- Σε αντίθετη περίπτωση το (b)



(a) $*(32, +(X, Y))$



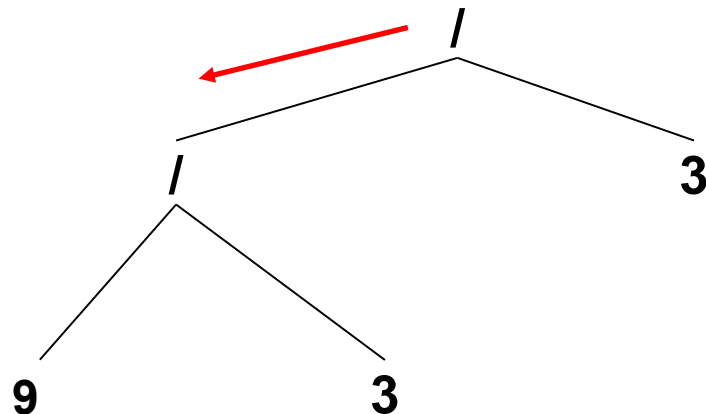
(b) $+(* (32, X), Y)$

Προσεταιριστικότητα

- Τι γίνεται σε περίπτωση που υπάρχουν δύο τελεστές ίσης προτεραιότητας?
- Για παράδειγμα:
9/3/3

Αριστερά προσεταιριστικοί τελεστές

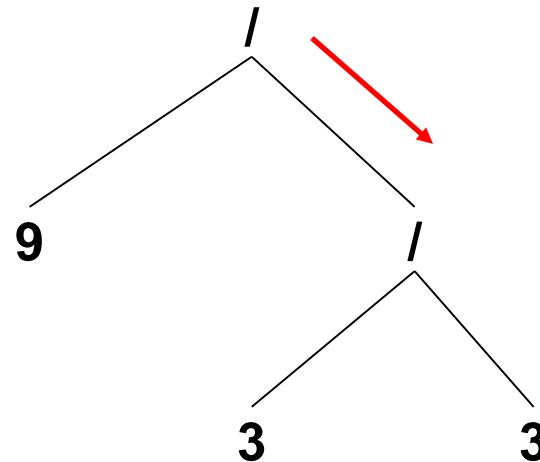
- Στα αριστερά του τελεστή υπάρχει μια έκφραση ίδιας ή μικρότερης τιμής σειράς.



Left-Associative $/(/(9,3),3)$

Δεξιά προσεταιριστικοί τελεστές

- Στα δεξιά του τελεστή υπάρχει μια έκφραση ίδιας ή μικρότερης τιμής σειράς.



Right Associative $/(9,/(3,3))$

Θέση τελεστών

■ Ενθεματικοί Τελεστές

- Εμφανίζονται ανάμεσα στα ορίσματα τους
- πχ. αριθμητικοί τελεστές

■ Προθεματικοί Τελεστές

- Μπροστά από τα ορίσματα
- Ο τελεστής `not` .

■ Επιθεματικοί Τελεστές

- Εμφανίζονται μετά τα ορίσματα
- πχ παραγοντικό $(23!)$.

Ορισμός τελεστών στην Prolog

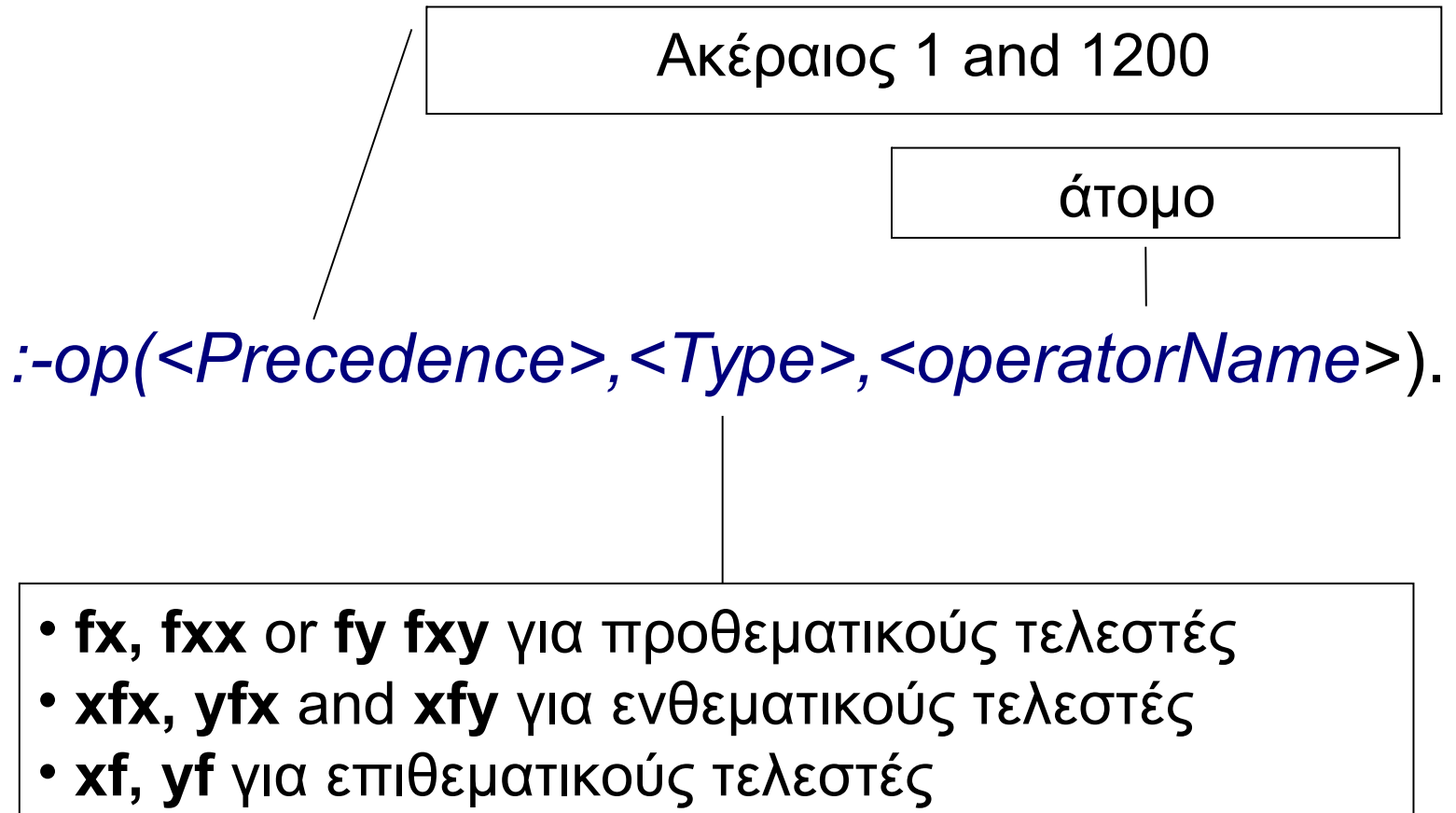
- Χρήση directives

- Κανόνες χωρίς κεφαλή (head) οι οποίοι βρίσκονται συνήθως στην αρχή του αρχείου.

- Ο ορισμός γίνεται με την χρήση του op/3 built-in κατηγορήματος:

`:-op(<Precedence>, <Type>, <operatorName>).`

Κατηγορία Op/3



Operator Type

- Type είναι ένα από τα
 - $xfx, xfy, yfx, yf, xf, fy, fx, fxx, fxy$
- f αναπαριστά τον τελεστή και x και y τα ορίσματα.
- x σημαίνει ότι η έκφραση στη συγκεκριμένη θέση πρέπει να έχει αυστηρά μικρότερη τιμή προτεραιότητας,
- y σημαίνει ότι ο τελεστής θα πρέπει να έχει μικρότερη ή το πολύ ίση τιμή προτεραιότητας.
- Για παράδειγμα:
 - yfx αριστερά προσεταιριστικός ενθεματικός τελεστής,
 - xfy δεξιά προσεταιριστικός ενθεματικός τελεστής.

Παραδείγματα και εσωτερική αναπαράσταση

■	<code>:- op(500,xfx,in).</code>	<code>A in B</code>	<code>in(A,B)</code>
■	<code>:- op(500,xfy,in).</code>	<code>A in B in C</code>	<code>in(A,in(B,C))</code>
■	<code>:- op(500,yfx,in).</code>	<code>A in B in C</code>	<code>in(in(A,B),C)</code>
■	<code>:- op(500,fx ,pre).</code>	<code>pre A</code>	<code>pre(A)</code>
■	<code>:- op(500,fy ,pre).</code>	<code>pre pre A</code>	<code>pre(pre(A))</code>
■	<code>:- op(500, xf,post).</code>	<code>A post</code>	<code>post(A)</code>
■	<code>:- op(500, yf,post).</code>	<code>A post post</code>	<code>post(post(A))</code>
■	<code>:- op(500, fxx,bin).</code>	<code>bin A B</code>	<code>bin(A,B)</code>
■	<code>:- op(500, fxy,bin).</code>	<code>bin A bin B C</code>	<code>bin(A,bin(B,C))</code>

Έλεγχος ορισμού Τελεστών

- *current_op/3* επιτρέπει να “δούμε” τους ορισμούς τελεστών.

current_op(<Prec>,<Type>,<OpName>).

- *write_canonical/1* εμφανίζει την εσωτερική μορφή των τελεστών.

- Κώδικας που εμφανίζει όλους τους ορισμένους τελεστές στη Prolog

*current_op(Prec,Type,OpName),
write(Prec), write(Type), write(OpName), nl,
fail.*

Εκφράσεις Προτασιακής Λογικής

- Αναπτύξτε ένα πρόγραμμα το οποίο πετυχαίνει αν η έκφραση της προτασιακής λογικής που δίνεται σαν ερώτηση στην Prolog είναι αληθής. Το πρόγραμμα θα πρέπει να αποτιμά εκφράσεις της μορφής:

?- t and f or f or t.

yes

Ο Ορισμός των Τελεστών

- Δύο τελεστές απαιτούνται
 - "and" και "or".
- Ενθεματικοί, αριστερά προσεταιριστικοί με προτεραιότητα στη σύζευξη.

`:-op(450,yfx,and).`

`:-op(500,yfx,or).`

Ορίζοντας τη σχέση and/2

and(Arg1,Arg2):-

Arg1,Arg2.

- Έφ'όσον έχουμε ορίσει τους τελεστές:

Arg1 and Arg2 :-

Arg1,Arg2.

Πλήρης Κώδικας (1/2)

%%% operator definitions.

:-op(450,yfx,and).

:-op(500,yfx,or).

%%% and operator definition

Arg1 and Arg2 :-

Arg1,Arg2.

Πλήρης Κώδικας (2/2)

%%% or operator definition

Arg1 or _Arg2 :-

Arg1.

_Arg1 or Arg2 :-

Arg2.

%%% truth values

t.

f:-!,fail.