

Εργαστήριο Λογικού Προγραμματισμού

Μανόλης Μαρακάκης, Καθηγητής

mmarak@cs.hmu.gr

**Τμήμα Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών
Σχολή Μηχανικών
Ελληνικό Μεσογειακό Πανεπιστήμιο**

Ενότητα 3: Μάθημα 5

□ 3. Αναδρομή, Λίστες και Αριθμητική σε Prolog

- 3.3. Αριθμητική σε Prolog
- 3.4 Προγραμματιστικές Τεχνικές
- 3.5. Παράδειγμα: Οι 8 - Βασίλισσες

3. Λίστες και Αριθμητική σε Prolog

3.3. Αριθμητική σε Prolog

□ Μερικοί από τους πιο σημαντικούς αριθμητικούς τελεστές:

Τελεστής

Πράξη

+	πρόσθεση
-	αφαίρεση
*	πολλαπλασιασμός
/	διαίρεση πραγματικών αριθμών
//	διαίρεση ακέραιων αριθμών
rem	μόδουλο (υπόλοιπο διαίρεσης ακεραίων)
floor(X)	Η τιμή είναι ο μεγαλύτερος ακέραιος ο οποίος είναι μικρότερος ή ίσος με τον X.
ceiling (X)	Η τιμή είναι ο μικρότερος ακέραιος ο οποίος είναι μεγαλύτερος ή ίσος με τον X.
round(X)	Η τιμή είναι ο πραγματικός ο οποίος είναι πλησιέστερα στο ακέραιο μέρος του X.
truncate(X)	Η τιμή είναι ο πλησιέστερος ακέραιος μεταξύ X και 0.

3. Λίστες και Αριθμητική σε Prolog

3.3. Αριθμητική σε Prolog

- ❑ Η Prolog έχει το δυαδικό τελεστή **is** για καταχώρηση αριθμητικών τιμών ή εκφράσεων σε μεταβλητές.
 - Η παράσταση δεξιά του **is** υπολογίζεται και το αποτέλεσμα καταχωρείται στην μεταβλητή αριστερά του **is**.
- ❑ Οι σχεσιακοί τελεστές που ακολουθούν συγκρίνουν αριθμητικές παραστάσεις υπολογίζοντας πρώτα τους τελεσταίους τους. Έστω ότι **E1** και **E2** είναι **βασικές (ground) αριθμητικές παραστάσεις**.
 - $E1 ::= E2$ Οι $E1$ και $E2$ έχουν ίσες αριθμητικές τιμές.
 - $E1 \neq E2$ Οι $E1$ και $E2$ δεν έχουν ίσες αριθμητικές τιμές.
 - $E1 < E2$ $E1$ είναι μικρότερο από το $E2$.
 - $E1 > E2$ $E1$ είναι μεγαλύτερο από το $E2$.
 - $E1 \leq E2$ $E1$ είναι μικρότερο ή ίσο από το $E2$.
 - $E1 \geq E2$ $E1$ είναι μεγαλύτερο ή ίσο από το $E2$.

3. Λίστες και Αριθμητική σε Prolog

3.3. Αριθμητική σε Prolog

□ Παράδειγμα 1

- ?- **X is** 8+7. Αυτός ο στόχος δίνει λύση $X = 15$.
- ?- **X+1 is** 8+7. Αυτός ο στόχος δεν είναι συντακτικά σωστός επειδή αριστερά του **is** πρέπει να υπάρχει μεταβλητή.
- ?- 1+4 **==** 2+3. Αυτός ο στόχος δίνει true.
- ?- 1+4 **==** 1+X. Αυτός ο στόχος δίνει μήνυμα λάθους για την δέσμευση της μεταβλητής X (Instantiation Error). **Η μεταβλητή X πρέπει να είναι δεσμευμένη με κάποια τιμή.**

3. Λίστες και Αριθμητική σε Prolog

3.3. Αριθμητική σε Prolog

- **Παράδειγμα 2:** Η σχέση $\text{length}(L, N)$ μετρά τα στοιχεία της λίστας για να βρει το μήκος της λίστας. Δύο περιπτώσεις διακρίνονται:
 - α) Εάν η λίστα είναι άδεια τότε το μήκος της είναι 0.
 - β) Εάν η λίστα δεν είναι άδεια τότε το μήκος είναι ίσο με το μήκος της ουράς συν ένα.

- **A τρόπος: Κατασκευή δομής (Συσσώρευση) στην κεφαλή πρότασης.**

$\text{length}([], 0).$

$\text{length}([H|Tail], N) :- \text{length}(Tail, N1),$

$N \text{ is } N1 + 1.$

Πρόγραμμα 3.12: Υπολογισμός Μήκους Λίστας με Συσσώρευση στην Κεφαλή πρότασης

- **B τρόπος: Κατασκευή δομής (Συσσώρευση) στην σώμα πρότασης.**

$\text{length}(L, N) :- \text{length1}(L, 0, N).$

$\text{length1}([], M, M).$

$\text{length1}([H|Tail], M, N) :- M1 \text{ is } M + 1,$

$\text{length1}(Tail, M1, N).$

Πρόγραμμα 3.13: Υπολογισμός Μήκους Λίστας με Συσσώρευση στο Σώμα πρότασης

3. Λίστες και Αριθμητική σε Prolog

3.4. Προγραμματιστικές Τεχνικές

- ❑ Η λέξη **σχήμα** σημαίνει «η μορφή, η εξωτερική όψη ενός πράγματος ή η γραμμική παράσταση ενός αντικειμένου». Ουσιαστικά η λέξη «σχήμα» περιγράφει κάποια γενικά χαρακτηριστικά ενός αντικειμένου.
- ❑ Στο **προγραμματισμό** η λέξη «**σχήμα**» περιγράφει τα **γενικά χαρακτηριστικά ενός προγράμματος** τα οποία μπορεί να είναι **κοινά και σε άλλα προγράμματα**.
- ❑ Προγραμματισμός με χρήση σχημάτων προγραμμάτων σημαίνει ότι την διαδικασία κατασκευής ενός προγράμματος ο προγραμματιστής την εντάσσει σε μια ομάδα προγραμμάτων της οποίας **όλα τα προγράμματα έχουν ίδια βασική δομή (συντακτική ή σημασιολογική δομή)**.

3. Λίστες και Αριθμητική σε Prolog

3.4. Προγραμματιστικές Τεχνικές

- ❑ Τα σχήματα προγραμμάτων βοηθάνε ένα προγραμματιστή επειδή ο προγραμματιστής έχει ήδη στο νου του το σκελετό του προγράμματος που θα κατασκευάσει.
 - Απλά **χρειάζεται να εξειδικεύσει κάποια στοιχεία που αφορούν μόνο το πρόγραμμα του** και όχι τα υπόλοιπα προγράμματα της ομάδας στην οποία έχει εντάξει το δικό του.
 - Τα **λεπτομερή στοιχεία μπορεί να αφορούν**
 - ❖ **είτε ορίσματα σε στοιχειώδεις τύπους**
 - ❖ **ή επιπλέον στοιχειώδεις τύπους.**

3. Λίστες και Αριθμητική σε Prolog

3.4. Προγραμματιστικές Τεχνικές

- ❑ **Ορισμός :** *Μετα-γνώση (meta-knowledge)* είναι γνώση για την γνώση.
- ❑ Σε κάθε πεδίο προβλήματος με γνώση υπάρχει και μετα-γνώση.
- ❑ Τα **σχήματα** εμπεριέχουν **προγραμματιστική μετα-γνώση**. Αυτή την *μετα-γνώση* τη χρησιμοποιούν αποτελεσματικά οι έμπειροι προγραμματιστές.
- ❑ Στη Prolog αυτή η μετα-γνώση μπορεί να παρασταθεί εύκολα λόγω της συντακτικής απλότητας της Prolog. Αυτό ισχύει και για τις άλλες γλώσσες του Λογικού Προγραμματισμού

3. Λίστες και Αριθμητική σε Prolog

3.4. Προγραμματιστικές Τεχνικές

- ❑ Ορισμός: Ένα **σχήμα προγράμματος** ή **σχήμα** παριστάνει ένα σύνολο προγραμμάτων τα οποία έχουν κοινή μορφή.
- ❑ Ορισμός: **Στιγμιότυπο σχήματος** ή **(στιγμιότυπο)** ονομάζεται κάθε πρόγραμμα που δημιουργείται/κατασκευάζεται από κάποιο σχήμα δεσμεύοντας μεταβλητές του σχήματος.

3. Λίστες και Αριθμητική σε Prolog

3.4. Προγραμματιστικές Τεχνικές

- ❑ **Τεχνική: Κατασκευή δομής στην κεφαλή μιας πρότασης.**
- ❑ Σε αυτή τη προγραμματιστική τεχνική το **αποτέλεσμα συσσωρεύεται στη επαγωγική δομή** (ή τύπο) δεδομένων χρησιμοποιώντας την στοίβα της Prolog.
- ❑ Η επαγωγική δομή που δημιουργείται σε κάθε αναδρομική κλήση είναι **ημιτελής** και
 - δεν υπάρχει πρόσβαση στο τμήμα που έχει ήδη κατασκευαστεί,
 - ένα τμήμα της αναμένεται να ολοκληρωθεί από την αναδρομική επεξεργασία των υπόλοιπων στοιχείων της επαγωγικής δομής η οποία ελέγχει την αναδρομή.
- ❑ Όταν ολοκληρωθεί η αναδρομική επεξεργασία των στοιχείων της επαγωγικής δομής εισόδου,
 - η **Prolog συνθέτει το τελικό αποτέλεσμα** το οποίο και επιστρέφει μέσω της επαγωγικής δομής η οποία κατασκευάζεται.

3. Λίστες και Αριθμητική σε Prolog

3.4. Προγραμματιστικές Τεχνικές

- ❑ **Τεχνική:** **Κατασκευή δομής στην κεφαλή μιας πρότασης.**
- ❑ Τον σκελετό αυτής της τεχνικής μπορούμε να τον δούμε ως το Σχήμα Προγράμματος 3.1
- ❑ Κατηγορημα(**ΒάσηΔομήςΕλέγχου**,
ΒάσηΚατασκευαζόμενηςΔομής).
- ❑ Κατηγορημα(**ΔομήΕλέγχου**,
ΤελικήΚατασκευασμένηΔομή) :-
- ❑ Κατηγορημα(**ΜικρότερηΔομήΕλέγχου**,
ΚατασκευασμένοΜέροςΔομής),
- ❑ Κατασκευή(**ΚατασκευασμένοΜέροςΔομής**,
ΤελικήΚατασκευασμένηΔομή).

Σχήμα Προγράμματος 3.1: Κατασκευή δομής στη κεφαλή πρότασης.

3. Λίστες και Αριθμητική σε Prolog

3.4. Προγραμματιστικές Τεχνικές

- ❑ **Παράδειγμα 4: Α τρόπος:** Σβήσιμο πρώτης εμφάνισης με τη τεχνική της "κατασκευής δομής στη κεφαλή πρότασης".
- ❑ π_1 : delete(X, [X|Vasi], Vasi).
- ❑ π_2 : delete(X, [Y|Tail1], Teliki) :-
delete(X, Tail1, Meriki),
Teliki = [Y|Meriki].
- ❑ Κατηγ(ΒάσηΔΕ, ΒάσηΚΔ).
- ❑ Κατηγ(ΔΕ, ΤελικήΚΔ :-
- ❑ Κατηγ(ΜικρΔΕ, ΚατΜέροςΔ),
- ❑ Κατασκ(ΚατΜέροςΔ, ΤελικήΚΔ).

3. Λίστες και Αριθμητική σε Prolog

3.4. Προγραμματιστικές Τεχνικές

- ❑ **Παράδειγμα 5: Α τρόπος:** Σβήσιμο όλων των επαναλήψεων με τη τεχνική της «κατασκευής δομής στη κεφαλή πρότασης».
- π_1 : deleteAll(_, [], Vasi) :-
Vasi = [].
- ❑ Κατηγ(**ΒάσηΔΕ**, **ΒάσηΚΔ**).
- ❑ Κατηγ(**ΔΕ**, **ΤελικήΚΔ**) :-
- ❑ Κατηγ(**ΜικρΔΕ**, **ΚατΜέροςΔ**),
- ❑ Κατασκ(**ΚατΜέροςΔ**, **ΤελικήΚΔ**).
- π_2 : deleteAll(X, [X | Tail1], Teliki) :-
deleteAll(X, Tail1, Meriki),
Teliki = Meriki.
- π_3 : deleteAll(X, [Y | Tail1], Teliki) :-
deleteAll(X, Tail1, Meriki),
Teliki = [Y | Meriki]..

3. Λίστες και Αριθμητική σε Prolog

3.4. Προγραμματιστικές Τεχνικές

- ❑ **Τεχνική: Κατασκευή δομής στο σώμα πρότασης στην αναδρομή.**
- ❑ Στη προγραμματιστική τεχνική της «κατασκευή δομής στο σώμα μιας πρότασης στην αναδρομή»
 - **το αποτέλεσμα συσσωρεύεται σε νέα επαγωγική δομή** (ή τύπο) δεδομένων η οποία είναι ίδια με την επαγωγική δομή (ή τύπο) δεδομένων του ορίσματος του κατηγορήματος που θα επιστρέψει τα αποτελέσματα.
 - Τα νέα στοιχεία καταχωρούνται με τη σειρά δημιουργίας τους και είναι **προσπελάσιμα** κατά τη κατασκευή της δομής.
- ❑ Για να υλοποιηθεί αυτή η τεχνική χρειάζεται ένα νέο κατηγορήμα, «**ΝεοΚατηγορήμα/3**», το οποίο θα **έχει τουλάχιστον τρία ορίσματα** με τους εξής ρόλους

3. Λίστες και Αριθμητική σε Prolog

3.4. Προγραμματιστικές Τεχνικές

- ❑ **Τεχνική:** Κατασκευή δομής στο σώμα πρότασης στην αναδρομή.
- ❑ Το 1^ο όρισμα είναι η **δομή ελέγχου** με τα δεδομένα του προβλήματος.
- ❑ Το 2^ο όρισμα χρησιμοποιείται για την **κατασκευή της δομής, στο σώμα πρότασης** του νέου κατηγορήματος με επεξεργασία από την «**βάση προς τη κορυφή**».
- Το 2ο όρισμα, «**ΒάσηΚατασκευαζόμενηςΔομής**», στη κλήση του νέου κατηγορήματος στο αρχικό κατηγορήμα, θα έχει την **αρχική τιμή της δομής** η οποία θα είναι η βασική της περίπτωση.

3. Λίστες και Αριθμητική σε Prolog

3.4. Προγραμματιστικές Τεχνικές

- ❑ **Τεχνική:** Κατασκευή δομής στο σώμα πρότασης στην αναδρομή.
- ❑ Το 3ο όρισμα, «**ΤελικήΚατασκευασμένηΔομή**», θα επιτρέψει το αποτέλεσμα της δομής που κατασκευάστηκε.
- ❑ Η δομή **βρίσκεται** στη βάση της στοίβας της αναδρομής και **πρέπει να μεταφερθεί στη κορυφή** ώστε να περάσει το αποτέλεσμα στο αρχικό κατηγορημα για να μπορεί να χρησιμοποιηθεί στη συνέχεια.

3. Λίστες και Αριθμητική σε Prolog

3.4. Προγραμματιστικές Τεχνικές

- ❑ **Τεχνική:** Κατασκευή δομής στο σώμα πρότασης στην αναδρομή.
- ❑ Τον σκελετό αυτής της τεχνικής μπορούμε να τον δούμε ως το επόμενο Σχήμα Προγράμματος 3.1

3. Λίστες και Αριθμητική σε Prolog

3.4. Προγραμματιστικές Τεχνικές

- ❑ Κατηγορημα (ΔομήΕλέγχου, ΤελικήΚατασκευασμένηΔομή) :-
ΝέοΚατηγορημα(ΔομήΕλέγχου, ΒάσηΚατασκευαζόμενηςΔομής,
ΤελικήΚατασκευασμένηΔομή).
- ❑ ΝεοΚατηγορημα (ΒάσηΔομήςΕλέγχου,
ΚατασκευασμένοΜέροςΔομής,ΤελικήΚατασκευασμένηΔομή) :-
ΤελικήΚατασκευασμένηΔομή = ΚατασκευασμένοΜέροςΔομής.
- ❑ ΝεοΚατηγορημα(ΔομήΕλέγχου, ΚατασκευασμένοΜέροςΔομής,
ΤελικήΚατασκευασμένηΔομή) :-
Κατασκευή(ΚατασκευασμένοΜέροςΔομής,
ΝέοΚατασκευασμένοΜέροςΔομής),
ΝεοΚατηγορημα(ΜικρότερηΔομήΕλέγχου,
ΝέοΚατασκευασμένοΜέροςΔομής,
ΤελικήΚατασκευασμένηΔομή) .

Σχήμα Προγράμματος 3.2: Βασικό σχήμα προγράμματος "κατασκευή δομής στο σώμα πρότασης".

3. Λίστες και Αριθμητική σε Prolog

3.4. Προγραμματιστικές Τεχνικές

- ❑ **Παράδειγμα 4: Β τρόπος:** Σβήσιμο πρώτης εμφάνισης με τη τεχνική της "κατασκευή δομής στο σώμα πρότασης".
- ❑ π_1 : delete(X, L1, L2) :-
 Vasiki = [],
 neo_katig(X, L1,
 Vasiki, Teliki),
 Teliki = L2.
- ❑ Κατηγ (ΔE , ΤελικήΚατασκευασμένηΔομή) :-
 ΝέοΚατηγ(ΔομήΕλέγχου, ΒάσηΚΔ, ΤεΛΚΔ).
- ❑ ΝέοΚατηγ (ΒάσηΔΕ, ΚατΜέροςΔ, ΤεΛΚΔ) :-
 ΤεΛΚΔ = ΚατΜέροςΔ.
- ❑ ΝέοΚατηγ(ΔE , ΚατΜέροςΔ, ΤεΛΚΔ) :-
 Κατασκ(ΚατΜέροςΔ, ΝέοΚατΜέροςΔ),
 ΝέοΚατηγ(ΜικρΔΕ, ΝέοΚατΜέροςΔ, ΤεΛΚΔ)
- ❑ π_2 : neo_katig(X, [X|Tail1], Meriki, Teliki) :-
 append(Meriki, Tail1, Teliki).
- ❑ π_3 : neo_katig(X, [Y|Tail1], Meriki, Teliki) :-
 append(Meriki, [Y], Nea_meriki),
 neo_katig(X, Tail1, Nea_meriki, Teliki).

3. Λίστες και Αριθμητική σε Prolog

3.4. Προγραμματιστικές Τεχνικές

❑ **Παράδειγμα 5: Β τρόπος:** Σβήσιμο όλων των επαναλήψεων με τη τεχνική "κατασκευή δομής στο σώμα πρότασης".

➤ deleteAll(X, L1, L2) :-

Vasiki = [],

neo_katig(X, L1,

Vasiki, Teliki),

Teliki = L2.

❑ Κατηγ (ΔΕ, ΤελικήΚατασκευασμένηΔομή) :-

ΝέοΚατηγ(ΔομήΕλέγχου, ΒάσηΚΔ, ΤεΛΚΔ).

❑ ΝέοΚατηγ (ΒάσηΔΕ, ΚατΜέροςΔ, ΤεΛΚΔ) :-

ΤεΛΚΔ = ΚατΜέροςΔ.

❑ ΝέοΚατηγ(ΔΕ, ΚατΜέροςΔ, ΤεΛΚΔ) :-

Κατασκ(ΚατΜέροςΔ, ΝέοΚατΜέροςΔ),

ΝέοΚατηγ(ΜικρΔΕ, ΝέοΚατΜέροςΔ, ΤεΛΚΔ)

➤ neo_katig(_, [], Meriki, Teliki) :- Teliki = Meriki.

➤ neo_katig(X, [X|Tail1], Meriki, Teliki) :- Meriki = Nea_meriki,

neo_katog(X, Tail1, Nea_meriki, Teliki).

➤ neo_katig(X, [Y|Tail1], Meriki, Teliki) :-

append(Meriki, [Y], Nea_meriki),

neo_katig(X, Tail1, Nea_meriki, Teliki).

3. Λίστες και Αριθμητική σε Prolog

3.4. Προγραμματιστικές Τεχνικές

- ❑ **Παράδειγμα 45:** Γράψετε κατηγορημα `minElement(L, Min)` με `mode(+,?)` το οποίο είναι αληθές εάν το στοιχείο `Min` είναι το μικρότερο στοιχείο από τα στοιχεία της λίστας `L`. Θεωρήσατε ότι τα στοιχεία της λίστας είναι ακέραιοι.
- ❑ **Α τρόπος με τη τεχνική της "κατασκευή δομής στη κεφαλή πρότασης".**
$$\text{minElement}([M], M).$$
$$\text{minElement}([H|T], H) \text{ :- } \text{minElement}(T, M), H < M.$$
$$\text{minElement}([H|T], M) \text{ :- } \text{minElement}(T, M), H \geq M.$$
- ❑ **Β τρόπος με τη τεχνική της "κατασκευή δομής στο σώμα πρότασης".**
$$\text{minElement}([H|T], \text{Min}) \text{ :- } \text{minElement1}(T, H, \text{Min}).$$
$$\text{minElement1}([], M, M).$$
$$\text{minElement1}([H|T], M, \text{Min}) \text{ :- } H < M, \text{minElement1}(T, H, \text{Min}).$$
$$\text{minElement1}([H|T], M, \text{Min}) \text{ :- } H \geq M, \text{minElement1}(T, M, \text{Min}).$$

Τέλος Διάλεξης

Ευχαριστώ!

Ερωτήσεις;