

Εργαστήριο Λογικού Προγραμματισμού

Μανόλης Μαρακάκης, Καθηγητής

mmarak@cs.hmu.gr

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

Ενότητα 3: Μάθημα 3

□ 3. Αναδρομή, Λίστες και Αριθμητική σε Prolog

➤ 3.1. Αναδρομή

➤ 3.2. Λίστες

✓ 3.2.1. Δομή και Χρήση της Λίστα.

✓ 3.2.2. Παραδείγματα Προγραμμάτων με Λίστες

3. Λίστες και Αριθμητική σε Prolog

3.1. Αναδρομή

- Η Prolog χρησιμοποιεί δομές όπως η λίστα οι οποίες ορίζονται **επαγωγικά**. Ένα σύνολο στοιχείων Σ έχει **επαγωγικό (inductive)** ορισμό εάν μπορεί να οριστεί με μια από τις παρακάτω τρεις διακριτές προτάσεις:
 - 1. Η *βασική πρόταση ή βάση*.
 - 2. Η *αναδρομική (recursive) ή επαγωγική (inductive) πρόταση*.
 - 3. Η *ακραία (extremal) πρόταση*.

3. Λίστες και Αριθμητική σε Prolog

3.1. Αναδρομή

- ❑ 1. Η **βασική πρόταση (βάση)** του ορισμού καθιστά κάποια στοιχεία μέλη του συνόλου Σ . Η βασική πρόταση έχει διπλό σκοπό.
 - Να δείξει ότι το σύνολο Σ δεν είναι άδειο και
 - να καθορίσει τα βασικά στοιχεία με τα οποία θα κατασκευαστεί το υπόλοιπο τμήμα του συνόλου.

Παραδείγματα:

- Η **άδεια λίστα []** είναι μέλος του **συνόλου των λιστών** και αποτελεί τη βάση ορισμού τους.
- Το **0** είναι βάση ορισμού του **συνόλου των φυσικών αριθμών**.

3. Λίστες και Αριθμητική σε Prolog

3.1. Αναδρομή

- ❑ 2. Η επαγωγή (*induction*) ή επαγωγική (*inductive*) πρόταση του ορισμού **καθορίζει** τον τρόπο με τον οποίο στοιχεία του συνόλου Σ μπορούν να ενωθούν για να δημιουργηθούν νέα στοιχεία.
- ❑ Η επαγωγική πρόταση **περιγράφει** τις πράξεις οι οποίες πρέπει να εκτελεστούν σε στοιχεία του συνόλου Σ για να κατασκευαστούν νέα στοιχεία του συνόλου Σ .
 - Για την **λίστα**, **τα στοιχεία κατασκευάζονται από**
 - ❖ την **άδεια λίστα**,
 - ❖ την **ειδική συνάρτηση** «.» πληθικότητας 2 (ή με τον ειδικό συμβολισμό «|») και
 - ❖ όρους όπως **άτομα, ακέραιοι, πραγματικοί, σύνθετοι όροι**.
 - Μια **μη κενή λίστα** έχει τη μορφή
 - ❖ « **.(Όρος, Λίστα)** » ή « **[Όρος | Λίστα]** » όπου «**Όρος**» είναι οποιοσδήποτε όρος της Prolog και «**Λίστα**» είναι οποιαδήποτε λίστα.

3. Λίστες και Αριθμητική σε Prolog

3.1. Αναδρομή

□ 2. Η επαγωγή (*induction*) ή επαγωγική (*inductive*) πρόταση (συνέχεια)

➤ Για την λίστα ο **τελεστής «|»** συνδέει μια **οντότητα** (**ακέραιο, άτομο, πραγματικό, σύνθετο όρο**) με μια λίστα (ένα στοιχείο) και **δημιουργεί μια νέα λίστα** (ένα νέο στοιχείο). Παραδείγματα:

❖ Η λίστα **«[a]»** κατασκευάζεται από

- την κενή λίστα **«[]»**,
- τη συνάρτηση **«.»** (ή με τον ειδικό συμβολισμό **«|»**) και
- το στοιχείο **«a»**
- ως εξής **«.(a,[])= [a]»** ή **«[a|[]]= [a]»**.

❖ Παρόμοια, η λίστα **«[a,b]»** κατασκευάζεται από

- το στοιχείο **«a»**,
- τη συνάρτηση **«.»** (ή τον ειδικό συμβολισμό **«|»**) και
- τη λίστα **«[b]»**
- ως εξής: **«.(a,[b])=.(a,.(b,[]))= [a,b]»** ή
«[a|[b]]= [a|[b|[]]]= [a,b]».

❖ Από την λίστα **.(a,[b,c])** ή **[a|[b,c]]** προκύπτει η λίστα **[a,b,c]**.

3. Λίστες και Αριθμητική σε Prolog

3.1. Αναδρομή

- ❑ Η **ακραία (extremal) πρόταση** καθορίζει ότι εάν ένα στοιχείο δεν μπορεί να αποδειχθεί ότι είναι μέλος του συνόλου Σ εφαρμόζοντας πεπερασμένο αριθμό επαναλήψεων τις προτάσεις 1 και 2 τότε δεν είναι μέλος του συνόλου.
- ❑ «**Αναδρομικός (recursive)**» σημαίνει αυτοαναφορά. Ο αναδρομικός (recursive) ορισμός χρησιμοποιείται για να δηλώσει αυτό που ονομάζουμε **επαγωγικός (inductive)** ορισμός.
 - Δεν είναι όλοι οι αναδρομικοί ορισμοί και επαγωγικοί ενώ ένας επαγωγικός ορισμός είναι και αναδρομικός.
 - Συνεπώς, ο όρος «επαγωγικός» είναι υποκλάση του γενικότερου όρου «αναδρομικός».

3. Λίστες και Αριθμητική σε Prolog

3.1. Αναδρομή

- ❑ Ο όρος αναδρομικός (recursive) στην επιστήμη υπολογιστών σημαίνει **αυτοαναφορά** και είναι πιο γενικός όρος από τον όρο **επαγωγικός** (inductive).
- ❑ Στο **προγραμματισμό** μια αναδρομική διαδικασία καλεί το εαυτό της είτε ευθεως ή έμεσα. Αναδρομικές διαδικασίες βασίζονται σε αναδρομικούς ορισμούς.
- ❑ Εάν ένα υποπρόγραμμα βασίζεται σ' ένα επαγωγικό (inductive) ορισμό τότε τα τμήματα του υποπρογράμματος αντιστοιχούν με φυσικό τρόπο στη βασική πρόταση και στην επαγωγική πρόταση του ορισμού. Η επεξεργασία αναδρομικών δομών γίνεται με αναδρομικά προγράμματα.
- ❑ *Η Prolog δεν υποστηρίζει άλλη εντολή επανάληψης εκτός από την αναδρομή.*
- ❑ Για να γράψουμε ένα αναδρομικό πρόγραμμα πρέπει να χωρίσουμε το πρόβλημα σε περιπτώσεις οι οποίες μπορούν να ανήκουν σε μια από τις εξής ομάδες:
 - 1. Οι **βασικές περιπτώσεις** στις οποίες δεν έχουμε αναδρομή.
 - 2. Οι **γενικές περιπτώσεις** στις οποίες η λύση κατασκευάζεται από λύσεις απλούστερων περιπτώσεων του αρχικού προβλήματος.

3. Λίστες και Αριθμητική σε Prolog

3.1. Αναδρομή

- Ας υποθέσουμε ότι θέλουμε να γράψουμε ένα πρόγραμμα σε Prolog το οποίο θα υπολογίζει το παραγοντικό ενός αριθμού $n \geq 0$. Ο αναδρομικός ορισμός της συνάρτησης παραγοντικό έχει ως εξής:

- **Βασική Περίπτωση** $0! = f(0) = 1$

- **Γενική Περίπτωση** $n! = f(n) = (n-1)! \cdot n = f(n-1) \cdot n$ όπου $(n-1)! = f(n-1) = 1 \cdot 2 \cdot 3 \cdots (n-2) \cdot (n-1)$

- **A τρόπος: Κατασκευή Δομής στην κεφαλή της πρότασης.**

`factorial(0, 1).`

`factorial(N, F) :- N > 0, N1 is N-1, factorial(N1, F1), F is N * F1.`

Πρόγραμμα 3.1: Υπολογισμός παραγοντικού με κατασκευή δομής στην κεφαλή της πρότασης.

- **B τρόπος: Κατασκευή Δομής στο σώμα της πρότασης.**

`factorial(N, F) :- factorial1(N, N, F).`

`factorial1(0, 1, 1).`

`factorial1(1, Facc, Facc).`

`factorial1(N, Facc, F) :- N > 1, N1 is N - 1, NewFacc is Facc * N1,
factorial1(N1, NewFacc, F).`

Πρόγραμμα 3.2: Υπολογισμός παραγοντικού με κατασκευή δομής στο σώμα της πρότασης.

3. Λίστες και Αριθμητική σε Prolog

3.1. Αναδρομή: Στοίβα αναδρομικών κλήσεων

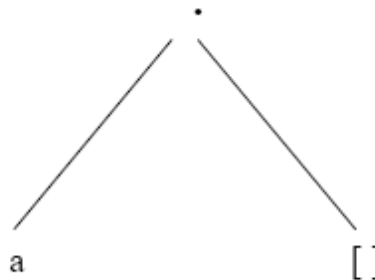
::	
$F1^3 = 1$	<code>factorial(0, F1³)</code>
$N1^3 = 0,$ <code>factorial(0, F1³)</code> $F1^2 = 1 * F1^3$	<code>factorial(1, F1²)</code>
$N1^2 = 1,$ <code>factorial(1, F1²)</code> $F1^1 = 2 * F1^2$	<code>factorial(2, F1¹)</code>
$N1^1 = 2,$ <code>factorial(2, F1¹)</code> $F = 3 * F1^1$	<code>factorial(3, F)</code>
Στοίβα	Κλήσεις

Σχήμα 3.1: Στοίβα αναδρομικών κλήσεων

3. Λίστες και Αριθμητική σε Prolog

3.2. Λίστες: Δομή και Χρήση της Λίστας

- ❑ Μια *λίστα* παριστάνεται σαν σύνθετος όρος με όνομα συνάρτησης την τελεία (.) και δύο ορίσματα, την κεφαλή και την ουρά της λίστας. Ο σύνθετος όρος `.(Κεφαλή, Ουρά)` παριστάνει μια λίστα με πρώτο στοιχείο την «Κεφαλή» και «Ουρά» την υπόλοιπη λίστα.
- ❑ Η ουρά η οποία είναι πάντα λίστα μπορεί να είναι είτε ένας σύνθετος όρος της ίδιας μορφής `.(Κεφαλή, Ουρά)` ή η άδεια λίστα.
- ❑ Λίστα με ένα στοιχείο παριστάνεται από την δομή `.(a, [])`. Αυτός ο όρος παριστάνεται από το δυαδικό δέντρο του σχήματος Σχήμα 3.1



Σχήμα 3.1: Δομή λίστας μ' ένα αντικείμενο

3. Λίστες και Αριθμητική σε Prolog

3.2. Λίστες: Δομή και Χρήση της Λίστας

- ❑ Οι πρωτεύουσες των νομών της περιφέρειας Κρήτης
➤ **chania, rethimnon, iraklion, agiosNikolaos**
μπορούν να γραφτούν ως όρος της Prolog ως εξής:
➤ `.(chania, .(rethimnon, .(iraklion, .(agiosNikolaos, []
))))`
- ❑ Για διευκόλυνση του προγραμματιστή η πιο πάνω
λίστα μπορεί να γραφτεί σ' ένα πρόγραμμα Prolog ως
εξής:
➤ `[chania, rethimnon, iraklion, agiosNikolaos]`
Εσωτερικά η Prolog αναπαριστά τις λίστες σαν
δυαδικά δέντρα.

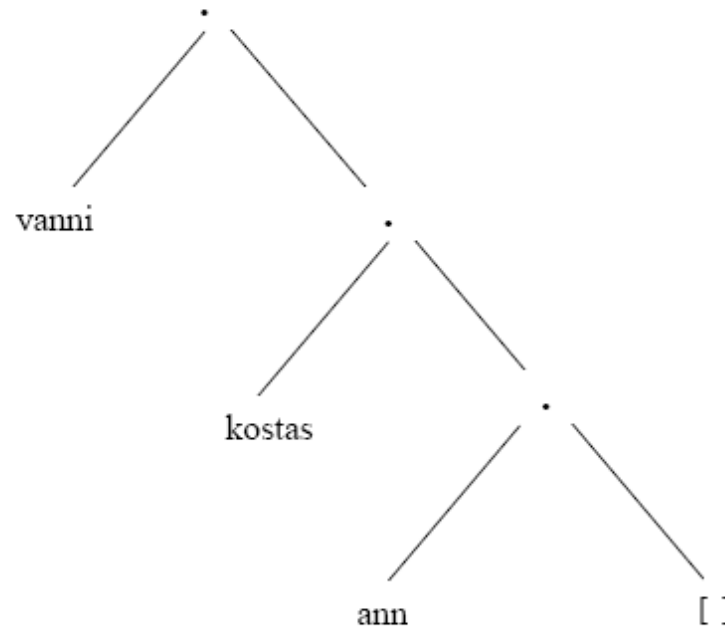
3. Λίστες και Αριθμητική σε Prolog

3.2. Λίστες: Δομή και Χρήση της Λίστας

- ❑ Έστω οι δύο τρόποι αναπαράστασης των λιστών,
 - α) σαν όρος της Prolog όπως η λίστα
 - ❖ `.(yannis, .(kostas, .(anna, [])))`
 - β) σαν μια λίστα στοιχείων μέσα σε αγκύλες όπως η λίστα
 - ❖ `[yannis, kostas, anna]`.
- ❑ Ο πρώτος συμβολισμός σαν όρος της Prolog είναι δυσνόητος ενώ ο δεύτερος συμβολισμός με τις αγκύλες είναι απλός και ευκρινής.
- ❑ Στο σχήμα Σχήμα 3.3 φαίνεται το δυαδικό δέντρο γι' αυτό το παράδειγμα.

3. Λίστες και Αριθμητική σε Prolog

3.2. Λίστες: Δομή και Χρήση της Λίστας



Σχήμα 3.3: Δομή της λίστας *[yannis,kostas,anna]*

3. Λίστες και Αριθμητική σε Prolog

3.2. Λίστες: Δομή και Χρήση της Λίστας

- ❑ Μια λίστα μπορεί να περιέχει σαν στοιχεία είτε άλλους όρους (ατομικούς ή σύνθετους), ή άλλες λίστες ή μεταβλητές. Δηλαδή, τα στοιχεία μιας λίστας μπορεί να έχουν την εξής μορφή.

- 1. σταθερές π.χ. yannis, 13
- 2. μεταβλητές π.χ. X, Sholi
- 3. σύνθετοι όροι π.χ. gonios(yannis, panos)
- 4. άλλες λίστες π.χ. [a, b, c, d]

❑ Παραδείγματα λιστών.

- [a, b, c, d]
- [[a, b], c, [d, e], f]
- [X, a, b, W, [Y, c]]
- [[a], [b, X]]

3. Λίστες και Αριθμητική σε Prolog

3.2. Λίστες: Δομή και Χρήση της Λίστας

- ❑ Στη λίστα [Head|Tail] η κάθετος (|) χωρίζει την κεφαλή από την ουρά. Σε λίστες που περιέχουν περισσότερα από ένα στοιχείο, η κεφαλή είναι το πρώτο στοιχείο της λίστας και η ουρά το υπόλοιπο μέρος το οποίο είναι λίστα. Στη λίστα [stef, sdo, seyr, steg] η κεφαλή είναι το άτομο stef ενώ η ουρά είναι η λίστα [sdo, seyr, steg].
- ❑ Εάν η λίστα περιέχει ένα μόνο στοιχείο, η κεφαλή είναι το μοναδικό στοιχείο της λίστας και η ουρά της είναι η κενή λίστα []. Για παράδειγμα, η λίστα [a] έχει για κεφαλή το a και ουρά το [].

❑ Παράδειγμα 1

	<u>Λίστα</u>	<u>Κεφαλή</u>	<u>Ουρά</u>
➤ α.	[a, b, c, d]	a	[b, c, d]
➤ β.	[[a, b], c, [d, e], f]	[a, b]	[c, [d, e], f]
➤ γ.	[X, a, b, W, [Y, c]]	X	[a, b, W, [Y, c],]
➤ δ.	[[a], [b, X]]	[a]	[[b, X]]

Τέλος Διάλεξης

Ευχαριστώ!

Ερωτήσεις;