

PROLOG

Λίστες

- Οι λίστες είναι η σημαντικότερη δομή δεδομένων της Prolog.
- Μια **λίστα** είναι ένας δυναμικός μονοδιάστατος πίνακας με στοιχεία του αριθμούς, σύμβολα, σύνθετους όρους ή και άλλες λίστες.
- Τα στοιχεία μιας λίστας δεν είναι απαραίτητο να είναι του ίδιου τύπου.
- Τα στοιχεία μιας λίστας τοποθετούνται μέσα σε αγκύλες [,] και χωρίζονται μεταξύ τους με κόμμα “ , ”.

Λίστες

- Παραδείγματα Λιστών σε Prolog
 - [maria, 8,ball, house].
 - [cat,point(1,2),[a ,b],d].
- Μια λίστα μπορεί να είναι:
 - Κενή (δομή χωρίς όρους) η οποία συμβολίζεται με []
 - Μια δομή με δύο όρους :την **κεφαλή (Head)** που είναι το πρώτο στοιχείο της λίστας και την **ουρά(tail)** που είναι το υπόλοιπο τμήμα της λίστας.

Λίστες

- Δύο **λίστες** ενοποιούνται εφόσον έχουν τον ίδιο αριθμό στοιχείων και τα αντίστοιχα στοιχεία τους μπορούν να ενοποιηθούν.

Λίστα 1	Λίστα 2	Ενοποίηση
[a, b, c]	[a,b,c,d]	αποτυχία
[a, b, c]	[X, Y]	αποτυχία
[a, b]	[a,X]	{ b / X }
[a,b]	[a X]	{ [b] / X }
[f(1), k, [m]]	[f(X), Y, Z]	{ 1/X, k/Y, [m] /Z }
[f(1), k, [m]]	[f(X), Y Z]	{ 1/X, k/Y, [[m]] /Z }
[[a]]	[X Y]	{ [a] / X, []/Y }

Λίστες

- Το κατηγορήμα `member` παρακάτω ελέγχει εάν ένα στοιχείο (πρώτο όρισμα) ανήκει σε μία λίστα (το δεύτερο όρισμα):
 - `member(X, [X | Y])`.
 - `member(X, [Head | Tail]) :- member(X, Tail)`.

Ερωτήσεις στην Prolog

?- `member(2, [1, 2, 3])`.

Yes

?- `member(4, [1, 2, 3])`.

No

?- `member(X, [1, 2, 3])`.

X=1 X=2 X=3

Λίστες

- Το παρακάτω κατηγορημα, $\text{last}(X,L)$ επιστρέφει ως X το τελευταίο στοιχείο της λίστας L .
 - $\text{last}(X, [X])$.
 - $\text{last}(X, [\text{Head} \mid \text{Tail}]) :- \text{last}(X, \text{Tail})$.
- Μπορούμε να κάνουμε κλήσεις στο $\text{last}/2$ με το πρώτο όρισμα ελεύθερο και το δεύτερο δεσμευμένο. Για παράδειγμα:
?- $\text{last}(Z, [2, 3, 4])$.
 $Z=4$
- Εάν καλέσουμε το κατηγορημα και με τα δύο ορίσματα δεσμευμένα, η απάντηση που θα πάρουμε θα είναι yes ή no, ανάλογα με το αν το πρώτο όρισμα είναι το τελευταίο στοιχείο της λίστας τους δεύτερου ορίσματος ή όχι

Λίστες

- Παραδείγματα:
?- last(4, [2, 3, 4]) .
Yes.
?- last(1, [2, 3, 4]) .
no.

Λίστες

- Η Συνένωση λιστών γίνεται με το κατηγορημα `append(L1,L2,L3)`, το οποίο επιτυγχάνει όταν η λίστα L3 είναι η ένωση (παράθεση) των L1 και L2.
- Η συνένωση μιας λίστας L2 με την κενή λίστα είναι η ίδια λίστα L2.
`append([], L2, L2)`.
- Η συνένωση μιας λίστας ενός στοιχείου και μιας λίστας L2 δίνεται από τον παρακάτω κανόνα:
`append([A], L2, [A|L2])`.

Λίστες

- Παραδείγματα

- `?-append([a,b],[c,d],[a,b,c,d]).`
`yes`

- `?-append([a,b,c],[1,2,3],L).`
`L=[a,b,c,1,2,3]`

- `?-append([a,[b,c],d],[a,[],b],L).`
`L=[a,[b,c],d,a,[],b]`

- `?-append(L1,L2,[a,b,c]).`
`L1=[], L2=[a,b,c]; L1=[a] L2=[b,c], L1=[a,b] L2=[c]; L1=[a,b,c] L2=[];`
`no`

Λίστες

- Η προσθήκη ενός στοιχείου στην αρχή μιας λίστας μπορεί να επιτευχθεί με τον ακόλουθο κανόνα
 - `add(List,Element,[Element|List]).`

Όπου

- `List` είναι η αρχική λίστα
- `Element` είναι το στοιχείο που θέλουμε να προσθέσουμε.
- `[Element|List]` είναι το αποτέλεσμα της προσθήκης.

Λίστες

- Η διαγραφή στοιχείων από μια λίστα γίνεται με το κατηγορημα `delete` όπου X είναι ένα στοιχείο, $L1$ μια λίστα και $L2$ η λίστα $L1$ από την οποία έχουν εμφανιστεί όλες οι εμφανίσεις του X .
- Οι βασικές ιδέες είναι οι εξής: Εάν η $L1$ είναι κενή, τότε και η $L2$ είναι κενή:
 - `delete(X, [] , [])`.
- Εάν το στοιχείο X ταυτίζεται με το πρώτο στοιχείο της $L1$, τότε η $L2$ ισούται με την ουρά της $L1$, από την οποία όμως έχουν αφαιρεθεί όλα τα X .
- `delete(X, [X | T1], L2) :- delete(X, T1, L2)`.

Λίστες

- Θέλουμε να ορίσουμε το κατηγορημα $\text{reverse}(L1, L2)$, το οποίο επιτυγχάνει όταν η λίστα $L1$ είναι η ανάστροφη της $L2$.
- Οι βασικές ιδέες είναι οι εξής:
- Η ανάστροφη της κενής λίστας είναι η κενή λίστα.
 - $\text{reverse}([], [])$.
- Για να αναστρέψουμε την $L1$, ξεχωρίζουμε το πρώτο στοιχείο της $H1$ από την ουρά της $T1$, αναστρέφουμε την ουρά της και έστω $RT1$ η ανεστραμμένη ουρά. Τότε η $L2$ ισούται με την ένωση της $RT2$ και του στοιχείου $H1$.
- $\text{reverse}([H1 | T1], L2)$:- $\text{reverse}(T1, RT1), \text{append}(RT1, [H1], L2)$.

Λίστες

- Παραδείγματα

?-reverse([1,2,3,4],[4,3,1,2]).

No

?-reverse([1,2,3,4],[4,3,2,1]).

Yes

?-reverse([a,b,c,d],L).

L=[d, c, b, a].